

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

OPTIMALIZAČNÍ TECHNIKY V OBRAZOVÝCH APLIKACÍCH

OPTIMIZATION TECHNIQUES IN IMAGE APPLICATIONS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Ondráček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Odstrčilík, Ph.D.

BRNO 2021

Diplomová práce

magisterský navazující studijní program **Biomedicínské inženýrství a bioinformatika**

Ústav biomedicínského inženýrství

Student: Bc. Pavel Ondráček

ID: 191744

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Optimalizační techniky v obrazových aplikacích

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši v oblasti optimalizačních a evolučních algoritmů vhodných pro zpracování obrazů. 2) Vybrané metody si blíže nastudujte a implementujte v programovém prostředí MATLAB. 3) Navrhněte vhodnou metodiku pro vzájemné srovnání a hodnocení vybraných optimalizačních algoritmů z pohledu různých aplikací ve zpracování obrazu a proveďte základní otestování. 4) Proveďte testování a vyhodnocení implementovaných algoritmů pro různé aplikace. 5) Proveďte diskusi dosažených výsledků a zhodnoťte účinnost a využitelnost aplikovaného řešení. 6) Vytvořené programové funkce přehledně okomentujte.

DOPORUČENÁ LITERATURA:

- [1] SIMON, D. Evolutionary optimization algorithms. Wiley, 2013, New Jersey, s. 742. ISBN 978-0-470-93741-9.
- [2] BIERLAIRE, M. Optimization: Principles and Algorithms. CRC Press, 2015, s. 718. ISBN 978-1-4822-0345-5.

Termín zadání: 8.2.2021

Termín odevzdání: 21.5.2021

Vedoucí práce: Ing. Jan Odstrčilík, Ph.D.

prof. Ing. Ivo Provazník, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Diplomová práce se zabývá způsoby optimalizace v obrazových aplikacích. Jsou zde popsány vybrané optimalizační metody a aplikace zpracování obrazu. Byly vybrány tři evoluční algoritmy, konkrétně včelí algoritmus, spojitý genetický algoritmus a algoritmus PSO, které byly podrobně popsány a implementovány v prostředí Matlab pro využití v registraci obrazu, přizpůsobené filtraci, segmentaci a rekonstrukci obrazu. Nechybí zde ani srovnání všech zrealizovaných algoritmů.

Klíčová slova

Optimalizace, evoluční algoritmy, včelí algoritmus, genetický algoritmus, algoritmus PSO, registrace obrazu, přizpůsobená filtrace, segmentace obrazu, rekonstrukce obrazu.

Abstract

This thesis deals with methods for optimization in image processing. There is described some of optimization techniques and some applications in image processing. There is also described detailed procedure and realization of bee algorithm, genetic algorithm, PSO algorithm and their realization in image registration, matched filtering, image segmentation and image reconstruction. Algorithms and their efficiencies are compared in the particular application for image processing.

Keywords

Optimization, evolution algorithms, bee algorithm, genetic algorithm, PSO algorithm, image registration, matched filtering, image segmentation, image reconstruction.

Bibliografická citace:

ONDRÁČEK, Pavel. *Optimalizační techniky v obrazových aplikacích*. Brno, 2021, 94 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství. Vedoucí práce Jan Odstrčilík.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Optimalizační techniky v obrazových aplikacích jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.“

V Brně dne: 20. května 2021

.....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Janu Odstrčilíkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: 20. května 2020

.....

podpis autora

Obsah

ÚVOD	12
1. OPTIMALIZACE A EVOLUČNÍ ALGORITMY.....	13
1.1 METODA NEJSTRMĚJŠÍHO SESTUPU	13
1.2 NEWTONOVA METODA.....	14
1.3 NÁHODNÉ PROHLEDÁVÁNÍ	14
1.4 SIMPLEXOVÁ METODA	14
1.5 SIMULOVANÉ ŽÍHÁNÍ	15
1.6 EVOLUČNÍ ALGORITMY (EA).....	15
1.6.1 Genetické algoritmy (GA)	15
1.6.2 Rojové algoritmy	17
1.6.3 Evoluční programování (EP)	20
2. OPTIMALIZACE V OBRAZOVÝCH APLIKACÍCH	21
2.1 REGISTRACE OBRAZU.....	21
2.1.1 Transformace obrazu	21
2.1.2 Interpolace.....	26
2.1.3 Kriteriační funkce	26
2.1.4 Registrační optimalizace	28
2.2 SEGMENTACE OBRAZU	28
2.2.1 Otsu metoda.....	28
2.2.2 Metoda narůstání oblastí.....	29
2.2.3 Metoda rozvodí	30
2.2.4 Segmentační optimalizace	30
2.3 PŘEDZPRACOVÁNÍ DAT PRO SEGMENTACI	30
2.3.1 Přizpůsobená filtrace.....	30
2.3.2 Optimalizace přizpůsobené filtrace	31
3. EVOLUČNÍ ALGORITMY V REGISTRACI OBRAZU	32
3.1 MODELOVÁ DATA.....	32
3.1.1 Včelí algoritmus.....	33
3.1.2 Spojitý genetický algoritmus	35
3.1.3 PSO	37
3.2 TESTOVÁNÍ NA FIRE DATABÁZI	39
3.2.1 Včelí algoritmus.....	41
3.2.2 Genetický algoritmus.....	45
3.2.3 PSO	48
3.2.4 Výsledky registrace.....	51
4. EVOLUČNÍ ALGORITMY V PŘIZPŮSOBENÉ FILTRACI	55
4.1 VČELÍ ALGORITMUS.....	57
4.2 GENETICKÝ ALGORITMUS	58

4.3	PSO.....	59
4.4	SROVNÁNÍ JEDNOTLIVÝCH ALGORITMŮ.....	59
5.	EVOLUČNÍ ALGORITMY V SEGMENTACI	62
5.1	VČELÍ ALGORITMUS.....	64
5.2	GENETICKÝ ALGORITMUS	65
5.3	PSO.....	66
5.4	SROVNÁNÍ JEDNOTLIVÝCH ALGORITMŮ.....	66
6.	EVOLUČNÍ ALGORITMY V REKONSTRUKCI OBRAZU	69
6.1	VČELÍ ALGORITMUS.....	69
6.2	GENETICKÝ ALGORITMUS	73
6.3	PSO.....	75
6.4	SROVNÁNÍ JEDNOTLIVÝCH ALGORITMŮ.....	77
7.	GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ	79
7.1	REGISTRACE OBRAZU.....	79
7.2	PŘÍZPŮSOBENÁ FILTRACE	80
7.3	SEGMENTACE OBRAZU	82
7.4	REKONSTRUKCE OBRAZU.....	83
ZÁVĚR		85

Seznam obrázků

Obr. 1: Jednobodové křížení. Upraveno z [27].	17
Obr. 2: Dvoubodové křížení. Upraveno z [27].	17
Obr. 3: Pohyb jedince PSO. Upraveno z [31].	19
Obr. 4: Grafický příklad translace.	22
Obr. 5: Grafický příklad rotace.	22
Obr. 6: Grafický příklad škálování.	23
Obr. 7: Grafický příklad zkosení.	24
Obr. 8 Grafický příklad afinní transformace.	24
Obr. 9: Referenční obraz a), pohybový obraz b).	32
Obr. 10: Závislost MSE na rotaci pohybového obrazu.	32
Obr. 11: Průběh hledání extrému včelího algoritmu.	33
Obr. 12: Průběh kvality včelího algoritmu.	34
Obr. 13: Průběh hledání extrému a průběh kvality genetického algoritmu.	36
Obr. 14: Průběh hledání extrému PSO algoritmu.	38
Obr. 15: Průběh kvality PSO algoritmu.	38
Obr. 16: Anotované body referenčního obrazu (vlevo) a pohybového obrazu (vpravo).	39
Obr. 17: Příklady snímků jednotlivých kategorií.	40
Obr. 18: Rozdělení snímku A07_01 do jednotlivých barevných složek.	41
Obr. 19: Závislost MSE a výpočetního času na počtu nejlepších včel.	43
Obr. 20: Závislost MSE a výpočetního času na počtu elitních včel.	43
Obr. 21: Závislost MSE a výpočetního času na počtu vyčkávajících včel.	44
Obr. 22: Závislost MSE a výpočetního času na počtu jedinců a iterací.	44
Obr. 23: Závislost MSE na pravděpodobnosti mutace.	46
Obr. 24: Závislost MSE na počtu elitních jedinců.	47
Obr. 25: Závislost MSE a výpočetního času na počtu jedinců a iterací.	48
Obr. 26: Závislost MSE na parametrech C_1 a C_2 .	49
Obr. 27: Graf závislosti MSE na parametru W .	50
Obr. 28: Závislost MSE a výpočetního času na počtu jedinců a iterací.	50
Obr. 29: Kategorie A – průměrné odchylky: $x=3,3$; $y=3,2$.	52
Obr. 30: a) kategorie B s průměrnými odchylkami: $x=4,3$; $y=33,5$, c) kategorie C s průměrnými odchylkami: $x=38$; $y=431$.	53

Obr. 31: Masky v jednotlivých směrech.....	56
Obr. 32: Parametrické obrazy získané pomocí jednotlivých masek.....	56
Obr. 33: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.....	57
Obr. 34: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.....	58
Obr. 35: Závislost Dice koeficientu a výpočetního času na počtu iterací a jedinců.....	59
Obr. 36: Fúze parametrických obrazů získaných na základě parametrů masky odvozené a) včelím algoritmem, b) genetickým algoritmem.....	60
Obr. 37: Segmentovaný snímek 10.jpg na základě: a) včelího algoritmu, b) genetického algoritmu, c) PSO algoritmu. d) zlatý standard.....	63
Obr. 38: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.....	64
Obr. 39: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.....	65
Obr. 40: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.....	66
Obr. 41: a) segmentovaný snímek 01.jpg, b) jeho zlatý standard, c) segmentovaný snímek 10.jpg, d) jeho zlatý standard.....	68
Obr. 42: Rekonstrukce binárního obrazu.....	69
Obr. 43: Vznik nových včel.....	71
Obr. 44: Závislosti kritériální funkce a času na počtu jedinců a iterací.....	71
Obr. 45: Proces rekonstrukce obrazu včelím algoritmem.....	72
Obr. 46: Vznik lepší poloviny chromozomů.....	73
Obr. 47: Závislosti kritériální funkce a času na počtu jedinců a iterací.....	74
Obr. 48: Proces rekonstrukce obrazu genetickým algoritmem.....	74
Obr. 49: Stanovení nejlepších pozic každého pixelu.....	75
Obr. 50: Závislosti kritériální funkce a výpočetního času na počtu jedinců a iterací.....	75
Obr. 51: Rekonstrukce obrazu algoritmem PSO.....	76
Obr. 52: Rekonstrukce výchozího obrazu na referenční (výchozí i referenční obraz převzat z [59]).	78
Obr. 53: Rekonstrukce výchozího obrazu na referenční, 2. ukázka (výchozí, resp. referenční obraz převzat z [64], resp. [65]).	78
Obr. 54: GUI blok – registrace.....	79
Obr. 55: GUI výběr oblasti pro výpočet kritériální funkce.....	80
Obr. 56: GUI blok – přizpůsobená filtrace.....	81
Obr. 57: GUI blok – segmentace obrazu.....	82
Obr. 58: GUI blok – výsledek segmentace.....	83
Obr. 59: GUI blok – výsledek segmentace 2. ukázka.....	83
Obr. 60: GUI blok – rekonstrukce obrazu.....	84

Seznam tabulek

Tabulka 1: Počáteční nastavení parametrů včelího algoritmu.....	42
Tabulka 2: Parametry počítače.....	42
Tabulka 3: Finální nastavení parametrů včelího algoritmu pro registraci.....	45
Tabulka 4: Počáteční nastavení parametrů genetického algoritmu.	46
Tabulka 5: Finální nastavení parametrů genetického algoritmu pro registraci.	48
Tabulka 6: Počáteční nastavení parametrů PSO.	48
Tabulka 7: Finální nastavení parametrů PSO pro registraci.	51
Tabulka 8: Rozřazení lícovaných snímků udávající počet párů spadajících do jednotlivých kategorií. Kategorie S, P a A udávají přítomnost anatomických rozdílů a míru překryvu snímků, viz kapitola 3.2.....	53
Tabulka 9: Srovnání algoritmů dle přesnosti a výpočetní náročnosti.	54
Tabulka 10: Finální nastavení včelího algoritmu pro přizpůsobenou filtraci.	58
Tabulka 11: Finální nastavení genetického algoritmu pro přizpůsobenou filtraci.	58
Tabulka 12: Finální nastavení parametrů PSO pro přizpůsobenou filtraci.	59
Tabulka 13: Parametry jednotlivých algoritmů.....	60
Tabulka 14: Vyhodnocení úspěšnosti segmentace.....	63
Tabulka 15: Nastavení parametrů včelího algoritmu pro segmentaci.	65
Tabulka 16: Nastavení parametrů genetického algoritmu pro segmentaci.	66
Tabulka 17: Nastavení parametrů algoritmu PSO pro segmentaci.	66
Tabulka 18: Kvantitativní hodnocení algoritmů a segmentace.	67
Tabulka 19: Kvantitativní hodnocení segmentace jiných autorů.	67
Tabulka 20: Nastavení parametrů včelího algoritmu pro rekonstrukci obrazu.	72
Tabulka 21: Nastavení parametrů genetického algoritmu pro rekonstrukci obrazu.....	74
Tabulka 22: Nastavení parametrů algoritmu PSO pro rekonstrukci obrazu.	76
Tabulka 23: Srovnání algoritmů v rekonstrukci obrazu.	77

ÚVOD

Optimalizace je technika, která je využívána k nalezení nejlepšího možného řešení v dané situaci v mnoha oborech, a dokonce i každým z nás. Optimalizujeme si denní rozvrh tak, aby průběh našeho dne byl co nejefektivnější. Stejně i v průmyslu se využívá k rozhodnutí, jakým způsobem transportovat zboží co nejrychleji a nejlevněji. S optimalizací se setkáváme také v medicíně, kde jsou snímány obrazy, ze kterých je dále stanovena diagnóza nebo plán léčby. Tyto obrazy je nutné optimalizovat takovým způsobem, aby byly lékaři snadno interpretovány.

Tato práce se zabývá problematikou optimalizačních metod a jejich využitím v obrazových aplikacích. Jsou zde uvedeny některé optimalizační techniky a vybrané aplikace zpracování obrazu. Podrobněji jsou popsány evoluční algoritmy, konkrétně včelí algoritmus, genetický algoritmus a PSO. Tyto algoritmy byly implementovány v programovacím jazyku Matlab pro využití v registraci obrazu. Pro testování jednotlivých algoritmů byla využita databáze FIRE, jež obsahuje 134 párů snímků sítnice.

Dalším využitím evolučních algoritmů může být segmentace dat, což bude dalším předmětem této práce společně s předzpracováním snímků pomocí přizpůsobené filtrace. K těmto aplikacím byly využity snímky sítnice z databáze DRIVE. Pomocí implementovaných evolučních algoritmů bude optimalizována hodnota prahu a parametry masky pro přizpůsobenou filtraci.

Poslední aplikací, ve které bude vyličen průběh evolučních algoritmů je rekonstrukce obrazu. Zde je přistoupeno k několika modifikacím pro korektní funkčnost uvedených algoritmů.

Algoritmy jsou v oblasti registrace porovnány na základě průměrné vzdálenosti referenčních bodů pohybového a referenčního obrazu. V přizpůsobené filtraci jsou hodnoceny na základě kritériální funkce, kterou je podobnostní koeficient Dice, jenž je také společně se senzitivitou, specificitou a přesností uplatněn ve srovnání algoritmů v segmentaci obrazu. Práce taktéž obsahuje srovnání s jinými autory, kteří pracovali se stejnou databází. V rekonstrukci obrazu jsou algoritmy srovnány na základě Euklidovy vzdálenosti referenčního a rekonstruovaného obrazu. Pro všechny aplikace je metrikou pro porovnání jednotlivých algoritmů také hodnota výpočetního času. Rovněž bylo vytvořeno grafické uživatelské rozhraní GUI pro vyšší přehlednost uživatele při testování vybrané aplikace.

1. OPTIMALIZACE A EVOLUČNÍ ALGORITMY

S optimalizací se setkáváme v nejrůznějších oborech, je uplatněna v ekonomických a biologických systémech, herních strategiích nebo v systémech zdravotní péče. Ve své podstatě se jedná o nalezení nejlepšího možného řešení vzhledem k určitému problému. Problém může mít vždy více ideálních řešení podle toho, z jakého hlediska je řešen. Proto je definována tzv. kritériální funkce, dle níž je vyhodnoceno nejlepší možné řešení, viz kapitola 2.1.3. Při optimalizaci jsou měněny stavové proměnné a na základě této změny je pozorována odezva celého systému. Cílem je se na základě iniciální hodnoty iteračně posunovat ke stále lepšímu řešení podle určeného pravidla do té doby, než je rozdíl mezi několika posledními úpravami nulový, nebo bylo dosaženo maximálního počtu iterací. Optimalizační problém se dá popsat jako minimalizace, či jako maximalizace problému. Někdy je zapotřebí najít minimum kritériální funkce, jindy naopak maximum kritériální funkce. Minimalizace se dá jednoduše konvertovat na maximalizaci a obráceně následujícím způsobem [7]:

$$\min_x f(x) \Leftrightarrow \max_x [-f(x)] , \quad (1)$$

$$\max_x f(x) \Leftrightarrow \min_x [-f(x)] . \quad (2)$$

Techniky řešící optimalizaci je možné rozdělit na několik typů a několik optimalizačních algoritmů, které budou blíže popsány v následující kapitole.

1.1 Metoda nejstrmějšího sestupu

Tato metoda je jedna z nejpobulárnějších algoritmů, které řeší optimalizaci a nejběžnější technika používaná k optimalizaci neuronových sítí. Je založena na postupné aktualizaci odhadu nejlepšího řešení (minima) pomocí numerického výpočtu derivací [8].

Nejprve je otestováno, zdali je odhad optimální. Jestliže ano, algoritmus končí, avšak zdali nikoliv, algoritmus pokračuje k dalšímu kroku, čímž je výpočet směru hledání, kterým se bude posouvat následující odhad. Směr hledání je vždy opačný ke směru gradientu účelové funkce, neboť je gradient směr nejstrmějšího růstu. Zároveň musí být určeno, s jakým krokem se bude optimum posouvat. Krok musí být určen kompromisem tak, aby bylo optimum nalezeno co nejrychleji a zároveň dané optimum nebylo překročeno. Jakmile je definován směr nového odhadu optima a krok, s jakým se bude pohybovat, je nový odhad optima aktualizován a algoritmus se vrací na začátek. Nevýhodou této metody – obecně metod založených na gradientu – je časté nalezení pouze lokálního minima [12, 21].

1.2 Newtonova metoda

Newtonova metoda se zakládá na geometrii. Jde o aproximaci účelové funkce kvadratickou funkcí v místě aktuálního odhadu optima, kde se následně zjistí minimum dané kvadratické funkce a zpřesní odhad optima. U této metody je potřebný výpočet Hessovy matice a parciálních derivací. Vztah používaný pro Newtonovu metodu lze popsat následující rovnicí:

$$[x_{n+1} \ y_{n+1}] = [x_n \ y_n] - (H_{[x,y]_n})^{-1} \nabla F_{[x,y]_n} , \quad (3)$$

kde $[x, y]_{n+1}$ definuje aproximaci řešení v aktuálním kroku, $[x, y]_n$ definuje aproximaci v předchozím kroku, $\nabla F_{[x,y]_n}$ definuje aproximaci první derivace v bodě $[x, y]_n$ a $H_{[x,y]_n}$ definuje numerickou aproximaci Hessovy matice v bodě $[x, y]_n$. Aby docházelo ke konvergenci, musí být Hessova matice pozitivně definitní. K tomu lze použít Sylvestrovo kritérium, jež říká, že matice je pozitivně definitní, pokud jsou determinanty submatic vyšší než nula. Jestliže Hessova matice není pozitivně definitní, má podobu jednotkové matice [2, 12].

1.3 Náhodné prohledávání

Tuto metodu lze považovat za jednu z nejjednodušších pro nalezení globálního extrému. Jde o algoritmus, jenž je založen na náhodě a generování pseudonáhodných čísel, proto se také označuje jako slepý algoritmus. Jde pouze o zapisování náhodně vygenerovaného odhadu tehdy, kdy je řešení lepší než v případě předchozího zapamatovaného odhadu optima [10].

1.4 Simplexová metoda

U simplexové metody je generován simplex S představující množinu nekomplanárních $d+1$ bodů (pro $d=3$ bude simplex S obsahovat 4 body). V simplexu je nalezen bod s nejvyšší x_h a nejnižší x_l funkční hodnotou, následně se vypočte těžiště z z bodů, z nichž se odstraní bod x_h . Přes dané těžiště se provede reflexe odstraněného bodu x_h , kde je definován nový bod y . Jestliže platí, že nově vytvořený bod vykazuje lepší řešení než bod x_h , bod y nahradí bod x_h . Pokud ovšem bod y shledává horší řešení nežli bod x_h , přistupuje se k redukci, jež má za následek smrštění všech bodů simplexu k bodu x_l [9, 25].

Popsaná metoda nabízí řadu modifikací, jako např. Nelderův-Meadův algoritmus. V této modifikaci je simplex definován jako konvexní útvar s $d+1$ vrcholy (pro $d=2$ je vytvořen trojúhelník). Po výpočtu těžiště ze všech bodů simplexu se provede transformace představující náhradu nejhoršího vrcholu simplexu za nový bod. Transformace, kromě redukce a reflexe, obsahuje také expanzi a kontrakci. Na základě několika pravidel se rozhoduje, ke které z těchto transformací bude přistoupeno [9, 25].

1.5 Simulované žíhání

Metoda je založena na fyzikálním principu žíhání oceli a spadá mezi nejúspěšnější tradiční stochastické optimalizační algoritmy. Při zahřátí kovu začnou atomy kmitat kolem svých počátečních souřadnic. S postupným ochlazováním kmitají čím dál méně a usazují se v místech s nižší energií. Kov je v daném případě definován vektorem x , jemuž se přidělí funkční hodnota (analogie k energii krystalové mřížky kovu). Žíháním se minimalizuje energie krystalu, což je v dané metodě simulováno jako minimalizace funkce. Z vektoru x se vytvoří vektor y , jenž nahradí původní vektor x s pravděpodobností určenou Metropolisovým vzorcem. Jestliže vektor y nabízí lepší či stejné řešení než vektor x , je vektor x nahrazen novým vektorem y s pravděpodobností 100 %. Pokud ovšem vektor y nabízí horší řešení než původní vektor y , je vektor x nahrazen vektorem y pouze s určitou pravděpodobností. V Metropolisově vzorci se vyskytuje teplota T určující pravděpodobnost akceptování nového vektoru, byť dává horší řešení. Na počátku je teplota T nastavena na vyšší teplotu a pravděpodobnost akceptování horšího vektoru je vyšší, s klesající teplotou se pravděpodobnost akceptování horšího řešení snižuje [7, 11].

Simulované žíhání spatřuje svůj benefit v úniku z lokálního optima a ve výsledku by metoda měla konvergovat ke globálnímu optimu. K dalším výhodám metody patří schopnost maximalizovat funkce, jež je jinak velmi obtížné nebo nemožné optimalizovat [11].

1.6 Evoluční algoritmy (EA)

Evoluční algoritmy jsou charakterizovány existencí populace jedinců, kteří jsou vystaveni tlaku životního prostředí, což vede k přírodnímu výběru nejlepšího jedince z populace. Obecně se evoluční algoritmy zabývají biologickými evolučními procesy [22].

1.6.1 Genetické algoritmy (GA)

Genetické algoritmy jsou nejdříve a nejvíce používané evoluční algoritmy. Jsou založeny na evoluci druhu a vychází z evoluční teorie Charlese Darwina. Tyto techniky byly původně vytvořeny spíše ke studování adaptivních systémů nežli jako prostředek k optimalizaci. GA jsou založeny na několika znacích přírodního výběru:

1. Biologický systém zahrnuje populaci jedinců, jež mají schopnost reprodukce.
2. Jedinci mají omezenou životnost.
3. V populaci existuje určitá odchylka.
4. Schopnost přežití je pozitivně korelována se schopností reprodukce.

GA simulují každý z těchto znaků přírodního výběru [7].

Pokud je znám optimalizační problém, je vytvořena populace s kandidáty nejlepšího řešení. Někteří kandidáti shledávají horší a někteří lepší řešení. Kandidáti s lepším řešením mají větší šanci na reprodukci. Rodiče zplodí potomky a následně jsou vyhozeni z populace, čímž vytvoří místo pro své potomky. Postupnou reprodukcí se stává populace silnější a silnější. Jedinci v populaci se nazývají chromozomy, které obsahují geny představující hledané parametry [22, 26].

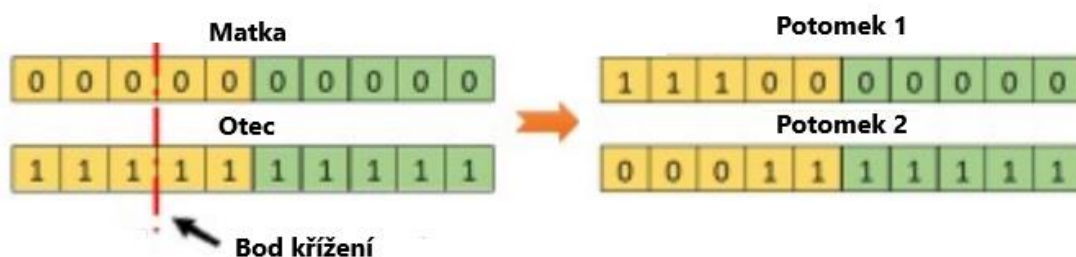
1.6.1.1 Spojitý genetický algoritmus

Počátek algoritmu začíná vygenerováním počáteční populace. Vzniklá populace je dána náhodnými hodnotami v příslušném rozsahu hledaných parametrů. Dále je pro každého jedince populace určena kvalita. Po určení kvality dochází k reprodukci jedinců (chromozomů). Dojde k seřazení chromozomů dle jejich kvality a dále se pracuje pouze s procentem nejsilnějších chromozomů. Poté dochází ke křížení chromozomů a vytváření nových potomků. Možností, jakými vybrat chromozomy ke křížení, je několik, např. náhodný výběr dvou chromozomů nebo párování shora dolů. Nový potomek je vytvořen určitou kombinací jeho rodičů. Pro křížení může být použita např. blending metoda, která bude popsána v kapitole 3.1.2. Po proběhnutí reprodukce se přistupuje k mutaci, což představuje náhodné změny genů v chromozomech. Mutace upravuje chromozomy o náhodně zvolené číslo a může k ní dojít pouze s určitou pravděpodobností. Takto reprodukováné a zmutované chromozomy vstupují do další iterace. U genetického algoritmu se vyskytuje také elitářství, jež spočívá v automatickém postupu nejlepšího chromozomu do další iterace. Algoritmus je ukončen v případě, že proběhl určitý počet iterací, nebo až se hodnota nejlepšího jedince po předem definovaném počtu iterací nebude měnit [23, 26]. Spojitý genetický algoritmus bude blíže popsán v kapitole 3.1.2.

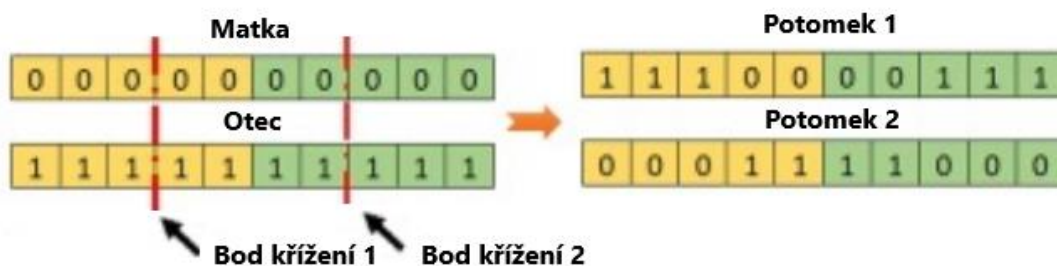
1.6.1.2 Binární genetický algoritmus

Jak již název napovídá, binární genetický algoritmus pracuje, na rozdíl od spojitého genetického algoritmu, pouze s binárními čísly. Nejprve je vygenerováno N sekvencí binárních čísel, kde N představuje počet chromozomů. Pro každý takový chromozom je vypočítána hodnota kritériální funkce. Dále postupuje pouze určité množství chromozomů, často bývá vybráno 50 % nejlepších jedinců. Další možností je postup všech chromozomů, splňujících určitý práh kvality. Tyto vybrané nejlepší chromozomy se podrobují křížení. Možností výběru chromozomů pro křížení je mnoho. Nejjednodušší metodou křížení je křížení shora dolů, kde se kříží vždy dva po sobě následující chromozomy. Sudé chromozomy mohou představovat matku, liché potom otce. Tento přístup je velmi jednoduchý, genetiku nicméně přesně nemodeluje. Další metodou je prostý náhodný výběr dvou různých chromozomů. Častou metodou pro selekci chromozomů pro křížení je tzv. vážená ruleta, v níž je určena pravděpodobnost výběru chromozomu dle jeho kvality a je spočítána distribuční funkce. Je vygenerováno náhodné

číslo v rozsahu 0-1 a určí se jedinec, který představuje interval distribuční funkce, do kterého náhodně generované číslo spadá. Po selekci chromozomů dochází k jejich křížení. Nejběžnější formou křížení je produkce dvou potomků ze dvou rodičů. Jednou z možností je jednobodové křížení, v němž jsou matka i otec rozděleni na dvě části náhodně vygenerovaným bodem křížení. Část genů otce dědí první potomek, druhou část druhý potomek. Stejným způsobem dědí potomci geny matky. Dalším způsobem je dvoubodové křížení, v němž jsou rodiče rozděleni na tři části pomocí dvou vygenerovaných bodů křížení. Jednobodové, resp. dvoubodové křížení, je zobrazeno na Obr. 1, resp. Obr. 2 [26].



Obr. 1: Jednobodové křížení. Upraveno z [27].



Obr. 2: Dvoubodové křížení. Upraveno z [27].

Takto křížením vzniklí potomci s určitou pravděpodobností podléhají mutaci. Každý bit, resp. gen, je s určitou pravděpodobností mutován z 0 na 1 a naopak.

Tímto způsobem je z populace rodičů vytvořena nová populace potomků. Stejně jako u spojitého genetického algoritmu je možnost elitářství, kdy do nové populace postupuje elitní jedinec ze staré populace. Z nově vzniklé populace potomků je vypočítána kritériální hodnota, stává se z ní populace rodičů a celý proces se opakuje [26].

1.6.2 Rojové algoritmy

Rojové algoritmy jsou inspirovány chováním živých organismů, jež žijí ve společenství a využívají emergence inteligentního chování roje. Vychází z toho, že spolu jedinci v roji komunikují. U těchto algoritmů je náhodně vytvořena populace jedinců, pro jednotlivé jedince je vypočtena funkční hodnota a je označen jedinec s nejlepší funkční hodnotou.

Ostatní jedinci tohoto nejlepšího jedince následují. U rojové inteligence se studují ptáci, včely, mravenci apod. Do nejvíce používaných rojových algoritmů lze zařadit algoritmus PSO, mravenčí kolonie či včelí algoritmus [14, 49].

1.6.2.1 PSO

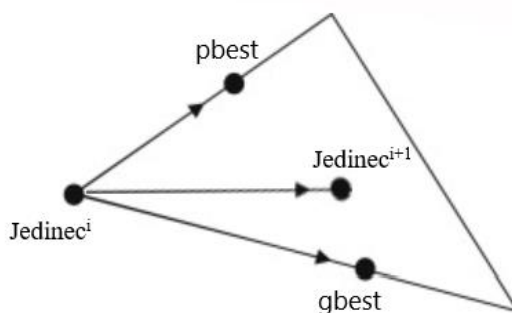
PSO (z angl. particle swarm optimization optimalizace rojem částic) je algoritmus navržen v 90. letech na základě studie, která ukázala, že některé skupiny zvířat, jimiž jsou ptáci nebo ryby, jsou schopny sdílet si navzájem informace, které vedou k přežití v přírodě. K vysvětlení algoritmu je představeno hejno ptáků, hledající místo k usazení, což představuje komplexní problém. Určení nejlepšího místa k usazení závisí na několika podmínkách. Jako jedna z podmínek může být maximum dostupné potravy nebo minimální šance na výskyt predátora, který by mohl hejno ohrozit. Ptáci v hejnu krouží a vyhodnocují jednotlivá nalezená místa do té doby, než je určeno nejlepší místo k usazení a hejno přistane najednou. Z této skutečnosti vyplývá, že spolu ptáci v hejnu komunikují, jinak by každý pták pravděpodobně přistál na jiném místě v jiný čas. Studie poukazuje na fakt, že každý jedinec, kromě informace o svém nejlepším nalezeném místě, nese i informaci o nejlepším nalezeném místě celého hejna [28].

Každý jedinec v populaci je reprezentován svou pozicí a rychlostí. Na počátku algoritmu jsou náhodně vygenerovány pozice jedinců. Rychlost každého jedince bývá inicializována náhodně nebo definována nulovou hodnotou [29]. V každé iteraci se aktualizuje pozice a rychlost každého jedince. Rychlost jedince je upravována na základě dvou informací: dosud nejlepší dosažená pozice jedince a nejlepší pozice dosažená celým hejnem. Úpravu rychlosti a pozice lze definovat vztahy 4 a 5 [28].

$$v_i^{t+1} = W \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \cdot (gbest - x_i^t), \quad (4)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (5)$$

kde v_i^{t+1} je rychlost částice v aktuálním kroku, v_i^t je rychlost v předešlém kroku, x_i^{t+1} je pozice v aktuálním kroku, x_i^t je pozice v předešlém kroku, $pbest_i$ je dosud nejlepší nalezená pozice jedince, $gbest$ je nejlepší nalezená pozice celým hejnem. Parametry c_1 , c_2 , W jsou akcelerační koeficienty ovlivňující míru aktualizace. Parametr c_1 udává individuální vědomost jedinců a parametr c_2 udává sociální vědomost jedinců, přičemž parametry jsou nejčastěji voleny v rozmezí $<0,5; 3>$. Parametr W je volen v rozmezí $<0; 1>$ a ovlivňuje směr pohybu z předešlé iterace. Jestliže je voleno $W=1$, je částice plně ovlivněna směrem pohybu z předešlé iterace [28, 30]. Na Obr. 3 je znázorněn možný pohyb jedince do další iterace, nová pozice jedince bude v závislosti na akceleračních koeficientech uvnitř vytvořeného trojúhelníku. Algoritmus PSO bude detailně popsán v kapitole 3.1.3.



Obr. 3: Pohyb jedince PSO. Upraveno z [31].

1.6.2.2 Mravenčí kolonie

Mezi základní vlastnosti mravenců patří nepřímá komunikace na základě zanechávání feromonové stopy. Mravenci v roji komunikují tak inteligentně, že na jejich základě byl vytvořen optimalizační algoritmus. Mravenčí kolonie jsou založeny na tom, že se jedinci náhodně pohybují a hledají potravu. Jakmile potravu nějaký jedinec nalezne, vrací se do mraveniště. Od bodu nalezení až do bodu mraveniště ovšem na cestě zanechává feromonovou stopu, aby dal ostatním jedincům vědět, kde se potrava nachází. Jestliže jedinec narazí na feromonovou stopu, bude ji následovat pravděpodobněji, než aby pokračoval ve své původní cestě. Čím více mravenců se pohybuje mezi naleznutou potravou a mraveništěm, tím větší je feromonová stopa, čímž se zvyšuje šance, že jedinec bude tuto cestu následovat. Cílem mravenců je, aby jejich cesta za potravou byla co nejkratší. Mravenčí kolonie se především používají při hledání optimální cesty v grafu, využívají se tak např. pro řešení problému obchodního cestujícího – obchodní cestující se snaží navštívit několik měst s návratem do původního města tak, aby jeho cesta byla co nejkratší, přičemž jsou známy vzdálenosti jednotlivých měst [7, 14].

1.6.2.3 Včelí algoritmus

Kolonii včel lze vysvětlit jako jedno velké společenství, jež se dokáže rozšířit stovky kilometrů do několika směrů pro nalezení zdroje potravy. V zásadě by květy, poskytující více nektaru a pylu za menší spotřeby energie včely, měly obsahovat více včel nežli ostatní květy. Vzhledem k velikosti jedince je podobně jako v jiných koloniích sociálního hmyzu velmi důležitá samoorganizace. Proces shánění potravy začíná již v úlu, z něhož jsou vyslány včely průzkumnice, které náhodně prohledávají prostor. Když se včely, které našly potravu, vrátí do úlu, přesouvají se na taneční parket. Tanec, jenž následně včela předvede, je typický pro včelí kolonii a je esenciální pro komunikaci kolonie. Obsahuje tři důležité informace: směr, vzdálenost a kvalitu potravy. Informace každého průzkumného jedince je získána kolonií pouze z tanečního parketu. Po tanci se včela tanečnice, tj. průzkumnice, vrací za potravou nyní již s více včelami, které čekaly v úlu. To umožňuje kolonii rychle a efektivně sbírat potravu. Po sklizni včely monitorují hladinu potravy, daná informace bude předána pomocí dalšího tance po návratu včel do

úlu. Pokud je zdroj potravy stále dostačující, bude tato informace předána při tanci a na následující cestu za potravou bude vysláno více včel [15].

U samotného algoritmu jsou včely rozděleny na včely průzkumnice a vyčkávací včely. Průzkumnice jsou náhodně zasazeny do prostoru, v každé iteraci je vybráno m nejlepších včel a z těchto včel jsou vybrány elitní včely. Následně jsou tyto včely následovány vyčkávacími včelami. Elitní včely jsou následovány více včelami než včely neelitní. Tyto následovnice prohledávají okolí kolem elitní včely, což reprezentuje lokální prohledávání. Pokud najde lepší řešení než elitní včela, je toto řešení včelou aktualizováno. Nová generace obsahuje m nejlepších včel. Zbytek včel je náhodně doplněn, aby odpovídal velikosti předchozí populace [20]. Podrobněji bude tento algoritmus popsán v kapitole 3.1.1.

1.6.3 Evoluční programování (EP)

Evoluční programování vytváří populaci jedinců, avšak nezahrnuje rekombinaci. Nové generace jsou vytvořeny pouze na základě mutace. Nejdříve je náhodně vytvořena populace jedinců x_i s předpokladem, že funkční hodnota pro všechny x je vyšší nebo rovna nule. Plození potomků je znázorněno rovnicí 6.

$$x'_i = x_i + r_i \sqrt{\beta \cdot f(x_i) + \gamma}, \quad (6)$$

r_i je náhodně generovaný vektor, v němž každý prvek odpovídá Gaussovské distribuci s nulovou střední hodnotou a rozptylem 1, β a γ jsou ladící parametry EP. Rozptyl mutace populace x_i je dána $(\beta \cdot f(x_i) + \gamma)$. Jeli parametr β nulový, pak mají všichni jedinci stejný průměrný stupeň mutace. Pokud je β vyšší než nula, pak jedinec s nízkou funkční hodnotou nemutuje tolik jako jedinec s vysokou funkční hodnotou. Často se jako výchozí nastavení definuje $\beta=1$ a $\gamma=0$ [7].

2. OPTIMALIZACE V OBRAZOVÝCH APLIKACÍCH

Optimalizační techniky jsou také hojně využívány v digitálním zpracování obrazu. Jako nejvíce nabízející se optimalizační problém ve zpracování obrazu se jeví registrace obrazu neboli lícování. Kromě registrace se pomocí optimalizace řeší několik dalších obrazových aplikací, např. segmentační problémy [19], komprese obrazu [13, 18, 61], adaptivní filtrace [62] nebo zvýšení jasu a kontrastu obrazu [63].

2.1 Registrace obrazu

V oboru zdravotnictví je hojně užíváno pořizování snímků. Tyto obrazy se poté aplikují při stanovení diagnózy pacienta, plánování léčby nebo sledování nádoru. Pro pochopení vývoje nemoci se v rámci medicínských postupů pracuje s několika obrazy zachycenými v různých časových intervalech. Často se pro získání informací o nemoci snímají obrazy o různých modalitách. Pro sledování a vyvozování informací z těchto pořízených snímků je zapotřebí obrazy na sebe naskládat s co nejlepším překrytím. Pro tento proces se používá registrace obrazu. Registrace je jednou ze základních metod digitálního zpracování obrazu. Jde o ztotožnění dvou nebo více obrazů tak, aby si jejich vzájemné prostorové souřadnice odpovídaly. Jsou-li obrazy pořízeny v odlišných časových intervalech, ale pomocí stejné modality, mluvíme o monomodální registraci. Pokud jde o sloučení dvou obrazů jiné modality, jedná se o multimodální registraci. Při registraci jsou definovány dva typy obrazu, referenční, který se během celého procesu registrace nemění, a pohybový, jenž je iterativně transformován. Cílem je připodobnit pohybový obraz tomu referenčnímu, k čemuž slouží geometrická transformace obrazu [1, 4].

2.1.1 Transformace obrazu

Transformací obrazu se rozumí nalezení vhodné transformační funkce, jež vyobrazí souřadnice vstupního obrazu do souřadnic výstupního obrazu. Jedná se pouze o změnu souřadnic, ostatní informace obrazu zůstávají nezměněné. Transformační funkce je reprezentována transformační maticí, jejímž vynásobením se vstupním obrazem se získá výstupní obraz s novými souřadnicemi.

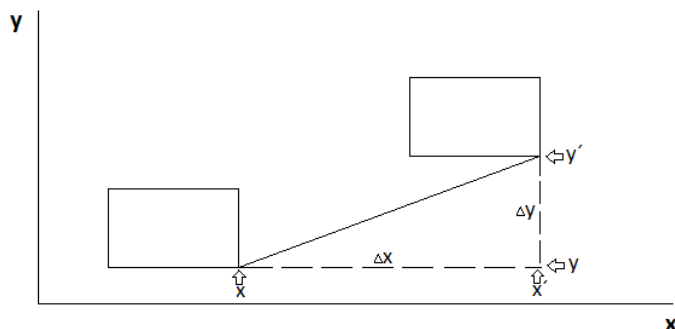
Geometrické transformace se rozdělují do dvou skupin, rigidní (tuhé) a flexibilní (pružné) transformace [3].

2.1.1.1 Rigidní transformace

Transformace, při nichž si části obrazu zachovávají tvar, rovnoběžnost a velikosti úhlů mezi objekty, se nazývají rigidní. Nejzákladnější rigidní transformací je translace (posun) a rotace (otočení) nebo kombinace obou zmíněných.

Translace – jedná se o prostý posun obrazu, jak lze vidět na Obr. 4. Translace se dá popsat následující rovnicí [5]:

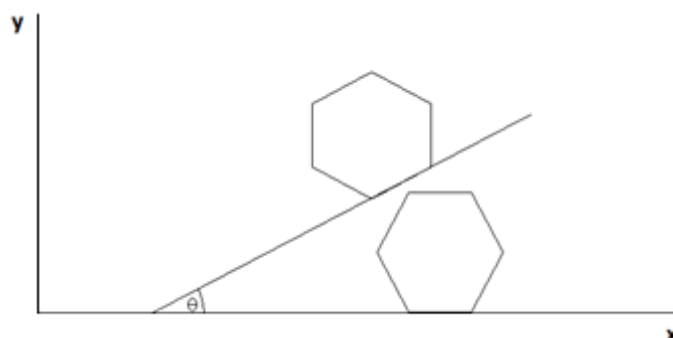
$$r' = r + \Delta r, \text{ kde } \Delta r = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \quad (7)$$



Obr. 4: Grafický příklad translace.

Rotace – jedná se o prosté otočení obrazu o zvolený úhel θ , jak lze vidět na Obr. 5. Rotace se dá popsat následující rovnicí [6]:

$$r' = B \cdot r, \text{ kde } B = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (8)$$



Obr. 5: Grafický příklad rotace.

Často se provádí rigidní transformace, v níž je užita translace i rotace současně. Pro tyto případy lze obě transformace sloučit, čímž je vytvořena obecná matice rigidní transformace R . V matici R jsou zahrnuty informace o rozdílu v posunu i velikosti úhlu. Obecnou rigidní transformaci lze definovat pomocí rovnice 9 [3].

$$r' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = R \cdot r = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta x \\ \sin \theta & \cos \theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (9)$$

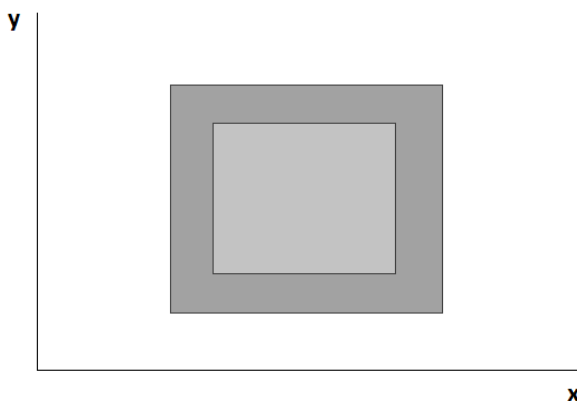
2.1.1.2 Flexibilní transformace

Flexibilní transformace se od rigidních liší tím, že mění vzdálenosti mezi pixely a způsobují deformaci struktur v obraze. Mezi základní flexibilní transformace patří škálování a zkosení [3].

Škálování – škálováním se rozumí změna měřítka struktur v obraze, viz Obr. 6, a dá se popsat transformační maticí S :

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

kde s_x představuje škálování v ose x a s_y v ose y [3].



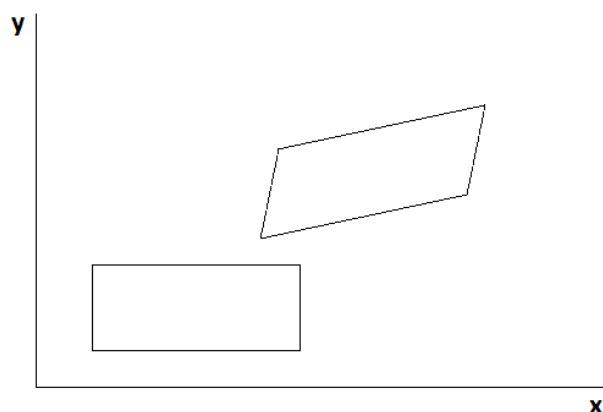
Obr. 6: Grafický příklad škálování.

Na Obr. 6 lze vidět tmavě šedý obrazec před a světle šedý obrazec po škálování, kdy došlo ke zmenšení měřítka v obou osách.

Zkosení – zkosení je postupně narůstající translace řádků a sloupců, viz Obr.7, a dá se popsat transformačními maticemi G_x a G_y [3].

$$G_x = \begin{bmatrix} 1 & g_{xy} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 0 & 0 \\ g_{yx} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (11)$$

kde G_x je transformační matice pro osu x a G_y je transformační matice pro osu y [3].



Obr. 7: Grafický příklad zkosení.

Na Obr. 7 je znázorněn příklad zkosení obrazce. Byla použita také translace pro lepší přehlednost.

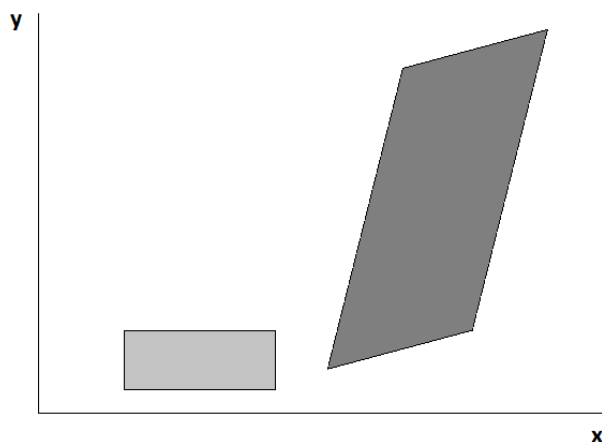
Jestliže jsou vynásobeny všechny výše zmíněné transformační matice, získáváme nejobecnější lineární transformaci – afinní transformaci [3].

2.1.1.3 Afinní transformace

Jak již bylo výše zmíněno, afinní transformace kombinuje dříve zmíněné transformace, čímž dochází k několika změnám, při nichž se ovšem nemění rovnoběžnost stran. Afinní transformační funkci lze popsat transformační maticí A [3]:

$$r' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = R \cdot S \cdot G_x \cdot G_y \cdot r = A \cdot r = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (12)$$

Na Obr. 8 je ukázán příklad vynásobené afinní transformační funkce s pixely obrazu, lze pozorovat, že výstupní obraz je posunut, rotován, škálován i zkosen. Světle šedý obrazec reprezentuje vstupní obraz a výstupní obraz je reprezentován tmavě šedým obrazcem.



Obr. 8 Grafický příklad afinní transformace.

Výše zmíněné transformace obsahují relativně málo parametrů, jež je třeba definovat. Existuje mnoho dalších, již složitějších transformací, které obsahují více parametrů, což má ale za následek obecnější transformaci. Jedná se např. o polynomiální transformaci, „piecewise“ transformaci nebo „thin-plate spline“ transformaci, která bude v následující kapitole blíže popsána [5].

2.1.1.4 Thin-plate spline (TPS)

Tato transformace využívající splajnové funkce je jednou z nejpoužívanějších transformací pro nerigidní registraci. Technika je založena na interpolaci a vyhlazování dat [5].

$$\{(x_i, y_i, f_i) : i = 1, \dots, N\}. \quad (13)$$

Je-li dáno několik bodů, jak je definováno v rovnici 13, je TPS interpolace definována následující rovnicí [5]:

$$f(x, y) = A_1 + A_2 \cdot x + A_3 \cdot y + \sum_{i=1}^N F_i \cdot r_i^2 \cdot \ln r_i^2, \quad (14)$$

kde $r_i^2 = (x - x_i)^2 + (y - y_i)^2 + d^2$. Zde je znázorněna rovnice desky nekonečného rozsahu deformace při zatížení umístěné v bodech $\{(x_i, y_i) : i = 1, \dots, N\}$. Deska se působením zatížení vychýlí, čímž získá hodnoty. Parametr d^2 se chová jako parametr tuhosti, s jeho hodnotou blížící se nule dochází více k bodovému zatížení a se vzrůstajícími hodnotami je vytvářen hladší povrch. Tento parametr tedy definuje tvar interpolačního povrchu a mění přesnost registrace. Rovnice 14 obsahuje $N+3$ neznámých. Dosazením souřadnic N bodů z rovnice 13 do rovnice 14, za předpokladu, že $f(x_i, y_i) = f_i$, je získáno N rovnic se třemi pravidly, viz rovnice 15, 16 a 17 [5].

$$\sum_{i=1}^N F_i = 0, \quad (15)$$

$$\sum_{i=1}^N x_i \cdot F_i = 0, \quad (16)$$

$$\sum_{i=1}^N y_i \cdot F_i = 0, \quad (17)$$

Rovnice 15 definuje takové pravidlo, že suma zatížení působících na desku je nulová. Toto pravidlo je nutné splnit, aby bylo zamezeno pohybu desky při vychylování a zůstala tak stacionární. Rovnice 16 a 17 vyžadují, aby momenty vzhledem k osám x a y byly nulové, což zajišťuje zamezení rotace desky pod zatížením [5].

2.1.2 Interpolace

Po provedení geometrické transformace se získají nové souřadnice pixelů. Velmi často ovšem dochází k tomu, že vlivem použití geometrické transformace nové souřadnice pixelů nezapadají do pravoúhlé mřížky vstupního obrazu. Proto je tyto body nutné interpolovat tak, aby do této mřížky zapadaly. Existuje několik interpolačních metod, níže budou představeny jen některé základní [3].

2.1.2.1 Metoda nejbližšího souseda

Tato metoda je nejjednodušší interpolační metodou a vyžaduje nejméně výpočetního času. Metoda uvažuje pouze jediný pixel, který je nejbližší interpolovanému bodu. Obraz interpolovaný touto metodou obsahuje nespojitosti a jeho spektrum je ovlivněno aliasingem. Na druhou stranu je tato metoda velmi rychlá, jelikož zahrnuje jen jeden pixel vstupující do interpolační funkce [3, 6, 16].

2.1.2.2 Bilineární interpolace

Bilineární interpolace uvažuje nejbližší 2x2 sousedství známých hodnot pixelů, které obklopují neznámý pixel. Následně je proveden vážený průměr těchto čtyř pixelů, čímž je získán interpolovaný pixel. Obraz interpolovaný touto metodou je následně hladší než obraz interpolovaný metodou nejbližším sousedem. Tato metoda na rozdíl od metody nejbližšího souseda není tak výrazně zasažena aliasingem, je však více výpočetně náročná [3, 16].

2.1.2.3 Bikubická interpolace

Bikubická interpolace je podobná interpolaci bilineární, uvažuje ovšem nejbližší 4x4 sousedství známých hodnot pixelů. Těchto 16 pixelů mají od neznámého pixelu rozdílnou vzdálenost, největší váhu obsahují pixely, které jsou neznámému pixelu nejbližší, nejvzdálenější pixely mají váhu nejnižší. Bikubickou interpolací je získáván ostřejší obraz v porovnání s předchozími metodami, to je však dáno vyšší výpočetní náročností [3, 16].

2.1.3 Kriteriaální funkce

K vyhodnocení úspěšnosti registrace je potřeba definovat kriteriaální funkci. Na základě zvolené kriteriaální funkce dochází k vyhodnocování míry podobnosti mezi referenčním a pohybovým obrazem. Při optimalizaci je cílem najít takové parametry, aby příslušná kriteriaální funkce byla co nejnižší. Existuje několik kriteriaálních funkcí, vybrané z nich budou níže popsány [3,17].

2.1.3.1 Euklidova vzdálenost

Společně s metodou nejmenších čtverců patří mezi nejjednodušší kritériální funkce. Je založená na absolutní velikosti dvou obrazů (vektorů). Tato metoda využívá rozdílů intenzit jednotlivých pixelů. Vztah pro euklidovskou vzdálenost je uveden v rovnici 18.

$$D_E(a, b) = |a - b| = \sqrt{\sum_{i=1}^N (a_i - b_i)^2}, \quad (18)$$

kde a je referenční obraz, b je pohybový obraz, i je jednotlivý index obrazu a N je počet prvků v obraze [24].

2.1.3.2 Metoda nejmenších čtverců

Metoda je velmi podobná euklidovské vzdálenosti, ovšem místo odmocnění sumy rozdílů je tato suma dělena počtem vzorků obrazu. Podobně jako u Euklidovské vzdálenosti je tato metoda založena na porovnání intenzit pixelů v obraze a je využívána u obrazů, jež vykazují vysokou podobnost. Vztah pro tuto metodu je uveden v rovnici 19.

$$SSD(a, b) = \frac{1}{N} \sum_{i=1}^N |a_i - b_i|^2, \quad (19)$$

kde a, b jsou obrazy, i je jednotlivý index obrazu, N je počet prvků v obraze [6, 17].

2.1.3.3 Korelační koeficient

Je využíván při lineárním vztahu intenzit pixelů v obraze. Vztah je zobrazen rovnicí 20.

$$CC(a, b) = \frac{\sum_i^N [a_i - \bar{a}] \cdot [b_i - \bar{b}]}{\sqrt{\sum_i^N [a_i - \bar{a}]^2 \cdot \sum_i^N [b_i - \bar{b}]^2}}, \quad (20)$$

kde \bar{a} je průměrná hodnota obrazu a , \bar{b} je průměrná hodnota obrazu b , i je jednotlivý index obrazu, N je počet prvků v obraze. Korelační koeficient nabývá hodnot $<-1, 1>$ a platí, že čím vyšší je hodnota korelačního koeficientu, tím lepší je registrace [6, 17].

2.1.3.4 Vzájemná informace

Vzájemná informace není závislá na hodnotách intenzit pixelů, avšak na vzájemném hodnocení pixelů. Metoda je založena na předpokladu, že intenzity obou obrazů představují náhodné veličiny. Je počítáno s mezními pravděpodobnostmi výskytu intenzit obrazů

a sdruženými pravděpodobnostmi mezi oběma obrazy. Vzájemnou informaci lze také

získat ze sdruženého histogramu. Podobně jako u korelačního koeficientu platí, že čím vyšší je hodnota vzájemné informace MI, tím je lepší registrace [6, 17].

2.1.4 Registrační optimalizace

Smyslem optimalizace v registraci obrazu je nalézt takové parametry transformační funkce, jež po vynásobení transformační funkce s pohybovým obrazem co nejvíce připodobňují referenční obraz. Mezi nejčastěji používané optimalizační techniky pro registraci obrazu patří simulované žíhání, náhodné kontrolované prohledávání a různé evoluční algoritmy [3].

2.2 Segmentace obrazu

Segmentace obrazu je jedním ze základních kroků analýzy obrazu. Je často užívána v oblasti zpracování medicínských dat a v široké míře se také používá v oblasti geografie a meteorologie. Výstupem tohoto procesu je obraz s ohraničenými oblastmi, které se navzájem nepřekrývají. Existuje několik metod pro segmentaci obrazu, jelikož má však každý obraz jiné pozadí a oblast zájmu, nelze s jistotou říct, která metoda se pro segmentaci individuálního obrazu hodí nejvíce. Nejčastěji používané jsou segmentace pomocí prahu, jenž je zvolen na základě informace v histogramu. Mezi tyto techniky se řadí metoda založená na maximální entropii a metoda minimálního vnitřního rozptylu mezi popředím (oblast zájmu) a pozadím, která bude v následující kapitole blíže popsána [32, 33].

Níže budou představeny některé segmentační přístupy. Přehled ostatních používaných segmentačních technik lze najít v [34] a [39].

2.2.1 Otsu metoda

Otsu metoda pojmenovaná po Nobuyuki Otsuovi je založená na váženém průměru rozptylů mezi třídami. Obraz je možné rozdělit na dvě třídy W_1 a W_2 viz vztah 21 a 22.

$$W_1 = \{0, 1, 2, \dots, T\}, \quad (21)$$

$$W_2 = \{T + 1, T + 2, \dots, L - 1\}, \quad (22)$$

kde T je práh rozdělení a L je celkový počet tónů šedi obrazu. Jestliže počet pixelů odpovídající hodnotě šedi i je n_i , celkový počet pixelů bude poté dán vztahem 23.

$$N = \sum_{i=0}^{L-1} n_i. \quad (23)$$

Pravděpodobnost výskytu hodnoty šedi je definována jako:

$$p_i = \frac{n_i}{N}, p_i \geq 0, \quad (24)$$

kde součet všech pravděpodobností výskytu musí odpovídat hodnotě 1. Pravděpodobnost obou tříd je:

$$P_{w1} = \sum_{i=0}^T p_i, \quad P_{w2} = 1 - P_{w1}. \quad (25)$$

Průměry tříd W_1 a W_2 lze vypočítat dle vztahů 26 a 27. Operací 28 je získán parametr sigma, který představuje hodnotu rozptylu mezi třídami. Optimální práh T je takový, při kterém sigma vykazuje maximální hodnotu [35].

$$\mu_{w1} = \sum_{i=0}^T \frac{i \cdot p_i}{P_{w1}}, \quad (26)$$

$$\mu_{w2} = \sum_{i=T+1}^{L-1} \frac{i \cdot p_i}{P_{w2}}, \quad (27)$$

$$\sigma^2 = P_{w1} \cdot P_{w2} \cdot (\mu_{w1} - \mu_{w2})^2. \quad (28)$$

Otsu metoda je jednoduchá technika, která je v praxi hojně využívána hlavně díky roli automatické detekce prahu, je však velmi citlivá na velikost šumu v obraze [35].

2.2.2 Metoda narůstání oblastí

Metoda narůstání oblastí je regionově orientovaný algoritmus, kdy je do obrazu zasazen jeden, nebo více startovacích bodů, tzv. semínek. Dále je určeno kritérium narůstání, např. jasová hodnota. Jestliže body sousedící se semínkem splňují kritérium, stávají se sami semínky. Tímto způsobem regiony semínek postupně narůstají až do bodu, kdy body okolo semínka nesplňují definované kritérium. Výsledek segmentace závisí na počáteční inicializaci semínek. Optimální počet semínek autoři v [36] zjišťovali na základě maximální entropie pomocí evoluční strategie. Kritérium může být statické, kdy se v průběhu narůstání nemění. V takovém případě je metoda náchylná vůči nerovnoměrnému osvětlení scény. Často se používá adaptivní kritérium, které se během narůstání mění. V úvahu se bere např. jen několik posledních hodnot nebo průměr hodnot již přidáných semínek do regionu [3, 37].

2.2.3 Metoda rozvodí

Metoda rozvodí je populární segmentační metoda, která má původ v oblasti matematické morfologie. K popisu metody bude sloužit analogie obrazu k topografickému reliéfu, kde výška každého bodu přímo souvisí s hodnotou jasu šedi. V případě, že začne pršet, je reliéf postupně zaplavován vodou a tvoří se „jezera“ v oblasti lokálních minim. Jezera jsou od sebe odděleny hrázemi, které chrání před slitím jezer dohromady. Obecně tato metoda není používána na originální obraz, nýbrž na jeho gradient. Velkou výhodou metody rozvodí je robustnost vůči šumu, na druhou stranu dochází k přesegmentování obrazu [3, 38].

2.2.4 Segmentační optimalizace

Smyslem optimalizace v segmentaci obrazu je nalézt takové parametry (práh apod.), které po segmentaci vstupního obrazu vykazují nejlepší výsledky (v případě dostupnosti anotovaných dat snaha o co nejpodobnější výstup segmentace s anotovanými daty, tzv. zlatým standardem). Mezi používané optimalizační techniky pro segmentaci patří např. PSO [43], genetický algoritmus [44] nebo optimalizace pomocí mravenčích kolonií [45].

2.3 Předzpracování dat pro segmentaci

Obrazy určené k segmentaci jsou často znehodnoceny šumem nebo nerovnoměrným osvětlením scény. V takovém případě je segmentace složitější a je nutné přistoupit k robustnějším přístupům segmentace nebo znehodnocená data předzpracovat, čímž dochází ke zvýšení úspěšnosti segmentačních metod. Takovým předzpracováním může být například ekvalizace histogramu pro zlepšení distribuce intenzit v histogramu, vážený mediánový filtr pro odstranění šumu „pepř a sůl“ často z CT snímků, Wienerův filtr pro restauraci zašuměného a rozmazaného obrazu a spoustu dalších [39].

V této diplomové práci bude podrobněji popsáno předzpracování obrazu pomocí přizpůsobené filtrace.

2.3.1 Přizpůsobená filtrace

Přizpůsobená filtrace je účinná technika pro detekci signálů vzniklá již v roce 1943 a její rozšíření na 2D je používáno např. k detekci cévního řečiště ve snímcích sítnice nebo zvýraznění otisku prstů. 1D přizpůsobený filtr je dán rovnicí 29.

$$g(m) = \sum_{i=1}^M f(i+m) \cdot h(i), \quad (29)$$

kde $f(i)$ je zašuměný signál a $h(i)$ je předpokládaný detekovaný signál. Rozšíření na 2D je potom dán rovnicí 30.

$$g(m, n) = \sum_{i=1}^M \sum_{j=1}^N f(i + m, j + n) \cdot h(i, j), \quad (30)$$

kde $f(i, j)$ je vstupní obraz, $h(i, j)$ je maska a M, N jsou její rozměry [40]. V prostorové oblasti lze filtraci definovat jako 2D konvoluci obrazu f s filtrační maskou h :

$$g(x, y) = f(x, y) * h(x, y), \quad (31)$$

kde x, y jsou prostorové souřadnice [41].

Základním krokem přizpůsobené filtrace je navržení masky. Maska je navržena na základě profilu objektu, který má být detekován. Je tedy nutná určitá apriorní znalost. Výstupem přizpůsobené filtrace je parametrický obraz, jehož odezva přímo závisí na míře korelace obrazu s navrženou maskou [54].

2.3.2 Optimalizace přizpůsobené filtrace

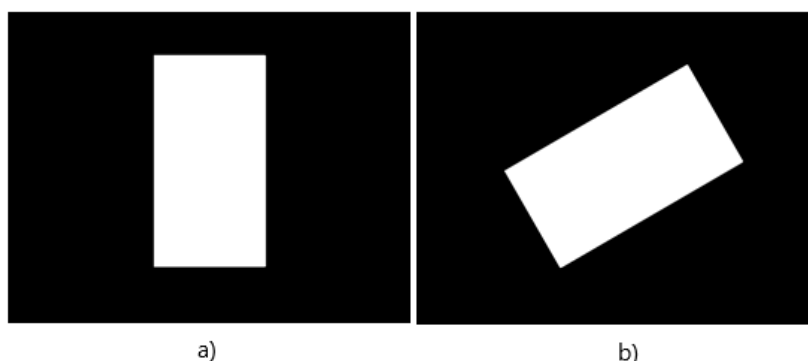
Odezva parametrického obrazu, která je výsledkem konvoluce filtrační masky a vstupního obrazu je závislá na parametrech navržené masky. Smyslem optimalizace u přizpůsobené filtrace je tedy optimalizovat parametry určující charakter filtrační masky takovým způsobem, aby se odezva parametrického obrazu co nejvíce přiblížila předpokládanému užitečnému signálu. Používané techniky jsou např. genetický algoritmus [46] nebo PSO [47].

3. EVOLUČNÍ ALGORITMY V REGISTRACI OBRAZU

Jak již bylo zmíněno, mezi jednu z nejčastějších obrazových aplikací, kde je využívána optimalizace, patří registrace obrazu. Optimalizačních technik, jež se dají pro tuto aplikaci využít, je mnoho. V této práci bylo přistoupeno na optimalizaci pomocí včelího algoritmu, spojitého genetického algoritmu a PSO.

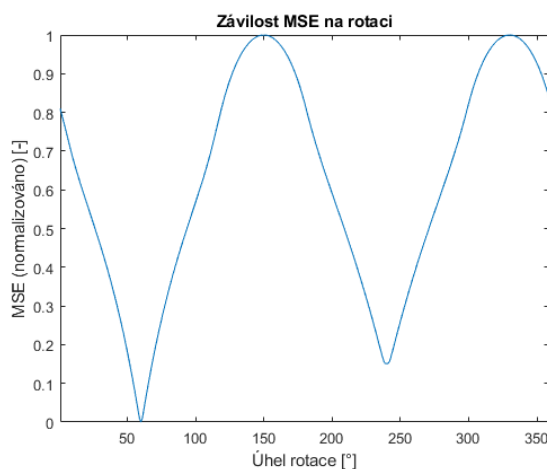
3.1 Modelová data

Pro pochopení jednotlivých algoritmů bylo testování nejprve provedeno na modelových datech. Byl vytvořen bílý obdélník na černém podkladě, který je brán jako referenční obraz. Dále byl vytvořen pohybový obraz, jenž je proti referenčnímu rotován o 60 stupňů. Referenční a pohybový obraz lze vidět na Obr. 9 a) a b).



Obr. 9: Referenční obraz a), pohybový obraz b).

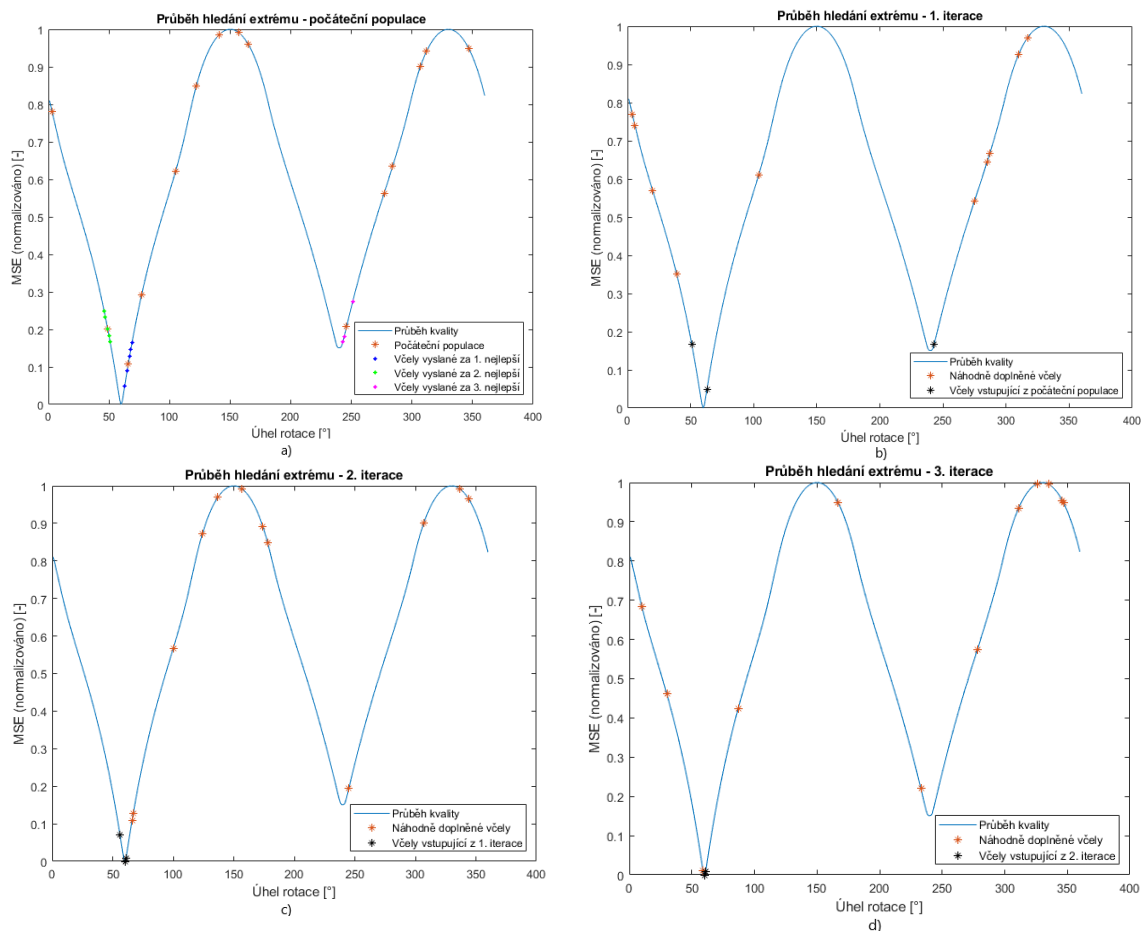
Jako kritériální funkce pro optimalizaci byla zvolena střední kvadratická odchylka (MSE), je tedy snaha o minimalizaci kritériální funkce. Na Obr. 10 je vyjádřena závislost hodnoty MSE na úhlu rotace pohybového obrazu vůči obrazu referenčnímu. Lze vidět, že minimální hodnotu MSE vykazuje rotace pohybového obrazu o 60 stupňů.



Obr. 10: Závislost MSE na rotaci pohybového obrazu.

3.1.1 Včelí algoritmus

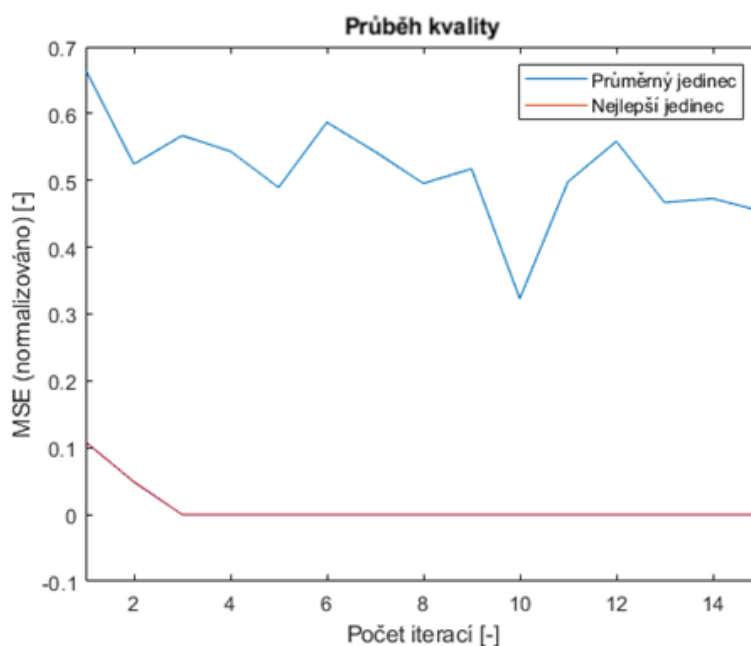
Prvním krokem včelího algoritmu je vygenerování počáteční populace v příslušném rozsahu. Je zvoleno 15 jedinců v populaci. Rozsah hledaného parametru θ je nastaven na $\langle 1; 360 \rangle$. Po vygenerování počáteční populace včel je určena kvalita každé z nich. Poté je populace seřazena od nejkvalitnější včely po nejméně kvalitní a je vybráno několik nejlepších včel, za kterými poletí další včely, aby prohledávaly lokální okolí. Pro ilustraci jsou zvoleny tři z nejlepších včel. Za nejlepšími včelami je vysláno několik včel tzv. vyčkávajících. Ty letí za kvalitnějšími včelami ve větším počtu než za méně kvalitními. Tyto kvalitnější včely se nazývají elitní včely a jejich počet je stanoven na dvě. Počet vyčkávajících včel je zvolen na tři a platí, že za elitními včelami jich letí dvakrát více než za včelami neelitními, tedy šest. Vyčkávající včely jsou zasazeny do lokálního prostoru včely, za kterou přiletěly. Nejkvalitnější vyčkávající včela potom tuto včelu nahrazuje. Jestliže ani jedna vyčkávající včela nevykazuje vyšší kvalitu, je včela zachována. Tímto způsobem jsou aktualizovány všechny nejlepší včely. Do další iterace postupují všechny nejlepší včely (buď aktualizované vyčkávajícími včelami, nebo stejné) a zbytek včel je náhodně vygenerován. Tímto se získá nová populace a proces se opakuje. Princip lze vidět na následujících obrázcích.



Obr. 11: Průběh hledání extrému včelího algoritmu.

Na Obr. 11 a) lze vidět počáteční populaci označenou červenou barvou. Z této počáteční populace jsou vybrány tři nejlepší včely. Jelikož je počet elitních včel nastaven na dvě, je za prvními dvěma nejlepšími včelami posláno dvakrát více vyčkávajících včel než za třetí, tedy šest. Za první nejlepší včelou jsou poslány včely označené modrou barvou. Tyto včely jsou náhodně zasazeny v prostoru v rozmezí ± 5 od nejlepší včely. Lze si všimnout, že dvě včely vykazují nižší hodnotu kritériální funkce, vykazují tedy vyšší kvalitu a lepší z nich nahrazuje nejlepší včelu, za kterou byla vyslána. Stejným způsobem operují včely znázorněné zelenou a růžovou barvou. Jediným rozdílem je, že počet vyčkávajících včel vyznačených růžovou barvou je menší, jelikož včela, za kterou byly vyslány, již není elitní.

Takto aktualizované včely postupují do další iterace, což lze vidět na Obr. 11 b), kde jsou zvýrazněny černou barvou. Náhodně vygenerovaný zbytek včel je vyznačen barvou červenou. Tato nová populace se podrobuje stejnému procesu jako počáteční populace a do další iterace postupují nové aktualizované včely. Tímto způsobem dochází ke stálému zvyšování kvalit včel, což lze vidět na Obr. 11 c) a Obr. 11 d). K ukončení algoritmu dochází často buď po určitém počtu iterací, nebo pokud je kvalita elitního jedince po několika iteracích neměnná, či rozdíl kvalit elitních jedinců menší než definovaná hodnota. Ukončovací kritérium pro tento případ je po proběhnutí definovaného počtu iterací, počet iterací je nastaven na 15. Na Obr. 12 lze vidět průběh kvality nejlepšího a průměrného jedince v každé iteraci. Lze si všimnout, že nejlepší jedinec se každou iterací zlepšuje, nebo zůstává na stejné hodnotě (řešení algoritmus našel ve třetí iteraci). Průměrný jedinec ovšem nevykazuje většího zlepšení, což je dáno náhodně doplněnými jedinci v každé iteraci.



Obr. 12: Průběh kvality včelího algoritmu.

3.1.2 Spojitý genetický algoritmus

Počátkem spojitého genetického algoritmu je vygenerování počátečních náhodných chromozomů ve zvoleném rozsahu, který je totožný jako u včelího algoritmu. Počáteční velikost populace odpovídá dvaceti jedincům. Pro všechny jedince je zjištěna jejich kvalita, dle níž jsou chromozomy seřazeny. Následně je vybrána lepší polovina jedinců, která podlehně křížení. Nejlepší počet jedinců je označen jako elitní jedinci, daní jedinci budou automaticky postupovat do další iterace, pro ilustraci je elitní počet nastaven na dva. Výběr rodičů pro křížení je vybrán náhodně. Křížení je uskutečněno pomocí blending metody. Blending metoda – metoda míchání chromozomů – kombinuje hodnoty obou rodičů za vytvoření nové hodnoty potomka. Z podstaty křížení vyplývá, že počet jedinců v populaci musí být sudý. Křížení probíhá na základě následujících vztahů [7]:

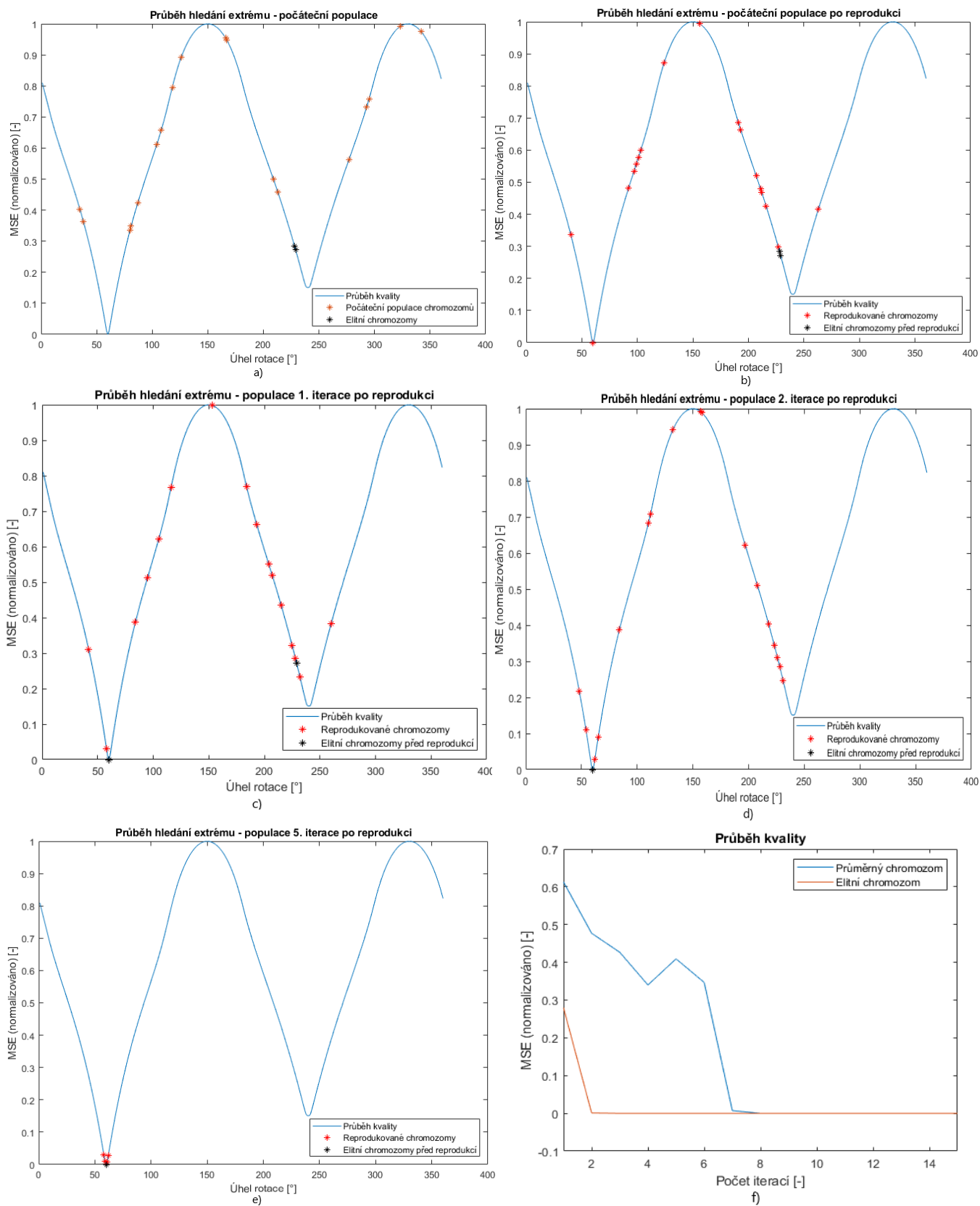
$$\begin{aligned}p_{n1} &= \beta p_{mi} + (1 - \beta) p_{oi}, \\p_{n2} &= \beta p_{oi} + (1 - \beta) p_{mi},\end{aligned}\tag{32}$$

kde p_{n1} , p_{n2} definují nové potomky, β definuje náhodné číslo v intervalu (0, 1), p_{mi} definuje gen s indexem i v chromozomu matky a p_{oi} definuje gen s indexem i v chromozomu otce. Po křížení podléhají potomci mutaci. Mutaci však podléhají pouze s určitou pravděpodobností, která byla stanovena na 40 %. Mutace je znázorněna rovnicí 33.

$$p_{nn} = p_n + (1 - k)(x - 0.5),\tag{33}$$

kde p_{nn} je potomek podléhající mutaci, p_n je nový potomek vzniklý křížením, x je náhodné číslo v intervalu (0, 1) a k je koeficient určující velikost mutace a odpovídá poměru čísla aktuální iterace a maximálního počtu iterací. Je tedy patrné, že se s každou iterací zmenšuje rozsah změny hodnoty chromozomu. Takto reprodukované a mutované chromozomy představují novou populaci chromozomů. První dva chromozomy nové populace jsou nahrazeny elitními jedinci a je vypočtena kvalita všech chromozomů, čímž začíná další iterace.

Na Obr. 13 a) je zobrazena počáteční populace chromozomů, pro všechny je spočítána kvalita a dva nejvyšší chromozomy jsou označeny jako elitní (černě označeni). Tyto chromozomy podléhají křížení a mutaci, reprodukovaná počáteční populace je vyobrazena na Obr. 13 b). Elitní jedinci označení před reprodukcí nyní nahrazují dva reprodukované chromozomy. Tato populace se stává novou generací pro další iteraci. Na obrázcích lze vidět, že se populace postupně „stahuje“ k elitním jedincům, viz Obr. 13 c) a d), až se nakonec přiblíží k optimálnímu řešení viz Obr. 13 e). K ukončení algoritmu dochází po maximálním počtu iterací, jenž je nastaven na 15.

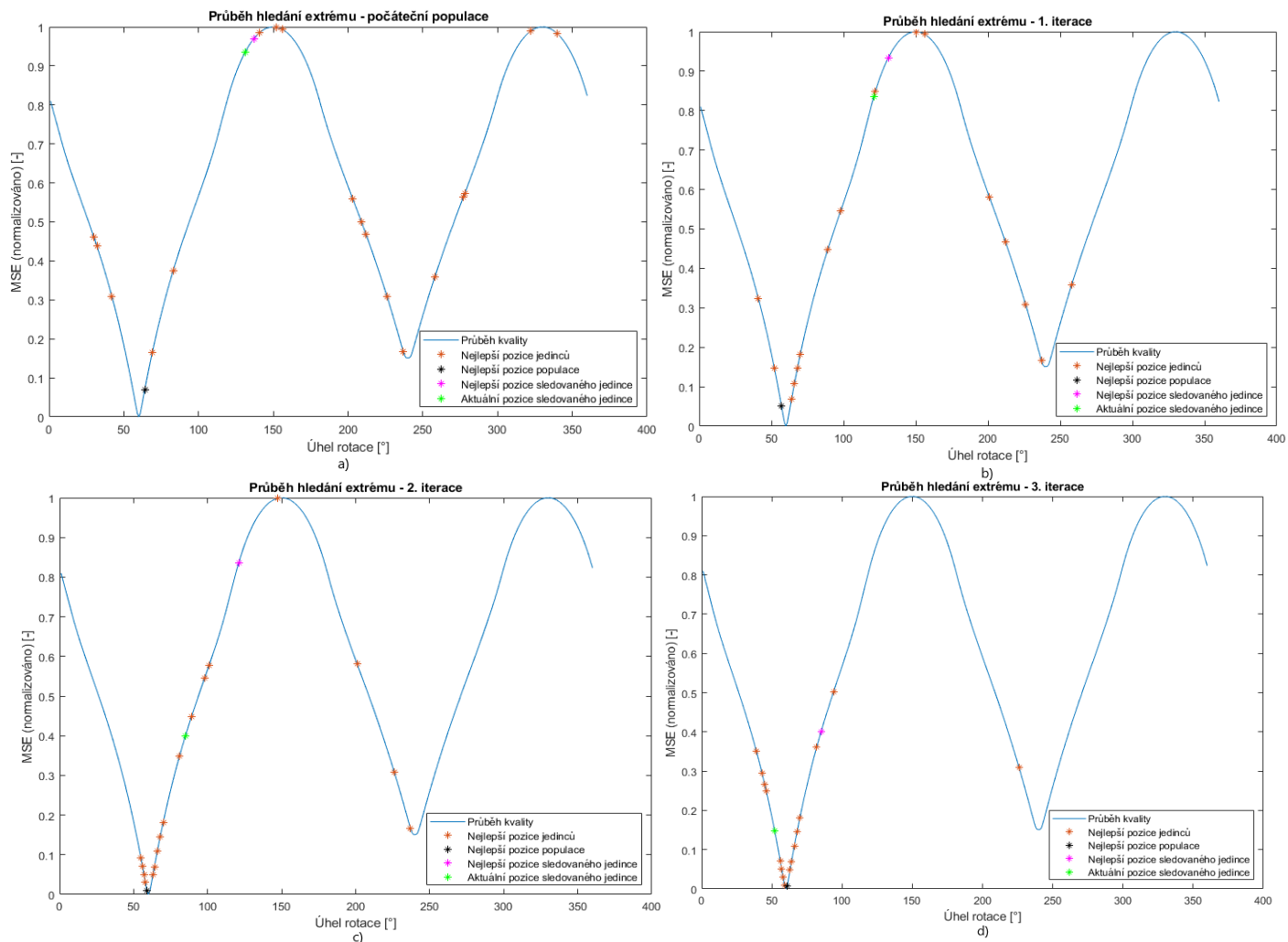


Obr. 13: Průběh hledání extrému a průběh kvality genetického algoritmu.

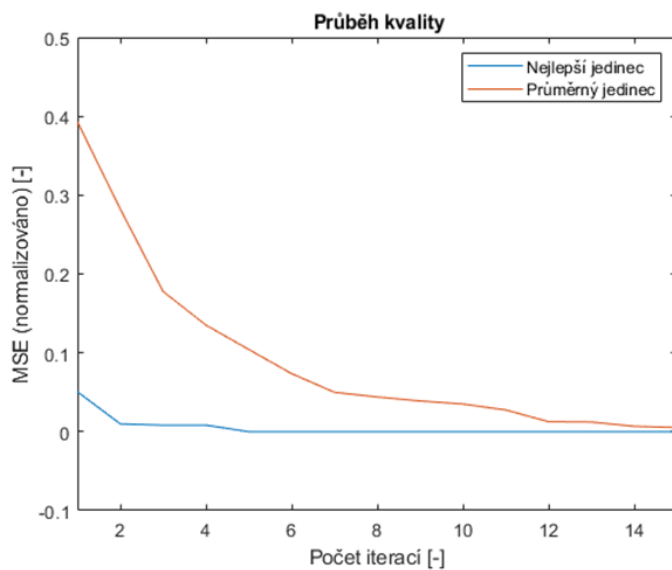
Na Obr. 13 f) lze vidět kvalitu průměrného a elitního chromozomu. Elitní chromozom se postupem iterací zkvalitňuje. Vlivem možného křížení elitního chromozomu s chromozomy neelitními dochází ke „stahování“ chromozomů směrem k elitnímu jedinci, čímž se zkvalitňuje hodnota většiny jedinců. Může se však stát, že křížením dvou chromozomů vznikne potomek, který vykazuje horší kvalitu, což lze vidět na Obr. 13 f) v paté iteraci. Nutno podotknout, že křížením pomocí blending metody vznikne potomek pouze v rozmezí hodnot obou rodičů.

3.1.3 PSO

Stejně jako u předchozích algoritmů je nejprve vygenerována počáteční populace, která činí dvacet jedinců. Zároveň s jedinci je vygenerována jejich počáteční rychlost. Počáteční rychlost všech jedinců je nastavena na nulovou hodnotu. Dále je nutné nastavit parametry c_1 , c_2 a W . Individuální parametr c_1 a sociální parametr c_2 jsou nastaveny na hodnotu 2. Parametr W je potom nastaven na 0,5. Pro všechny jedince je vypočtena jejich kvalita a je označen nejlepší jedinec v populaci. Pozice jedinců v počáteční populaci zároveň představují jejich dosavadní nejlepší pozici. Dle vztahu 4 a 5 je přepočítána rychlost a aktuální pozice každého jedince. Pro tyto aktuální pozice je vypočtena kvalita, jestliže vykazují vyšší kvalitu než jejich dosavadní nejlepší pozice, je jejich nejlepší pozice přepsána pozicí aktuální. Po takto aktualizované populaci je zjištěna nejlepší pozice celé populace, která je následně opět použita při přepočtu jednotlivých rychlostí. Pro ilustraci je vybrán jeden jedinec, jehož chování v každé iteraci lze vidět na následujících obrázcích. Na Obr. 14 a) je červeně vyznačena počáteční populace jedinců, jejichž pozice jsou zároveň i nejlepšími pozicemi. Růžovou barvou je vyznačen jedinec, jehož chování bude detailněji popsáno. Pro každého jedince je přepočítána rychlost, kterou ovlivňuje pozice nejlepšího jedince v populaci, který je vyznačen černou barvou. Dále je rychlost ovlivněna nejlepší pozicí, které jedinec dosáhl. V počáteční populaci je tato pozice zároveň aktuální pozicí. Na základě rychlosti je potom přepočítána aktuální pozice jedince, která je vyznačena zelenou barvou. Z Obr. 14 a) je patrné, že aktuální pozice jedince dosahuje vyšší kvality než jeho dosavadní nejlepší pozice. Nejlepší pozice je tedy aktualizována. Toto porovnávání se děje se všemi jedinci v populaci, následně je zjištěna nejlepší pozice populace – globální minimum – a začíná nová iterace. Na Obr. 14 b) lze vidět, že nejlepší pozice jedince je na místě aktuální pozice jedince z předchozí iterace. Nejlepší pozice celé populace je rovněž aktualizována. Tímto způsobem se jedinec přibližuje pozici nejlepšího jedince, což lze vidět na Obr. 14 c) a d). Průběh kvality nejlepšího a průměrného jedince v každé iteraci lze vidět na Obr. 15. Vykazuje podobné vlastnosti jako genetický algoritmus, kdy dochází ke zlepšování průměrného jedince, což je dáno ovlivněním pozic všech jedinců pozicí nejlepšího jedince v populaci (vyznačen černou barvou). Ukončovací mechanismus je obdobný jako u předchozích dvou algoritmů.



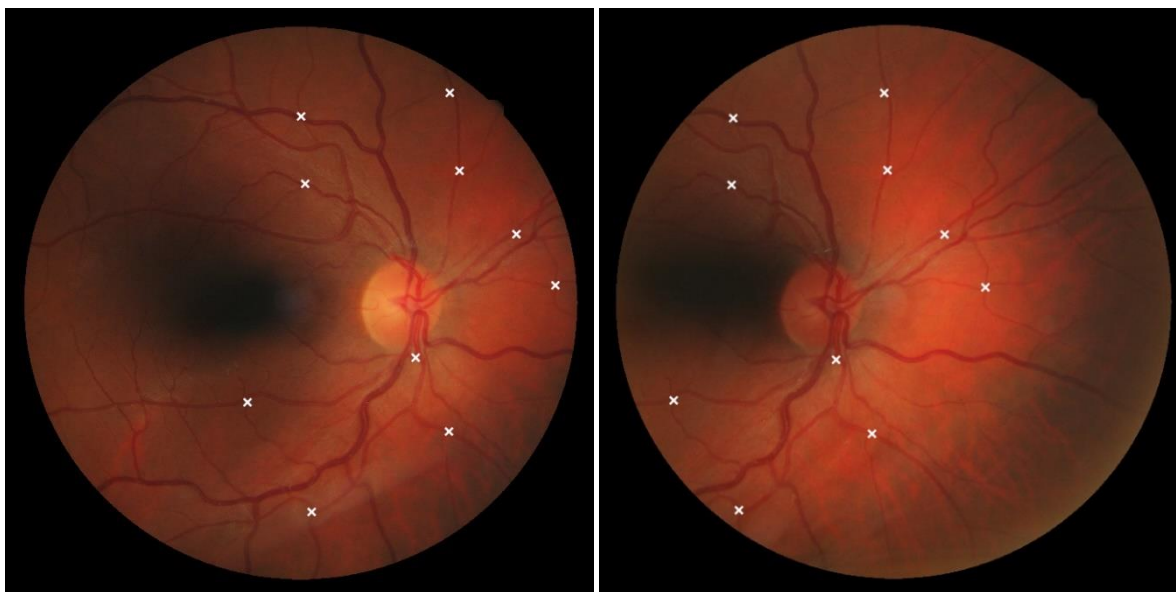
Obr. 14: Průběh hledání extrému PSO algoritmu.



Obr. 15: Průběh kvality PSO algoritmu.

3.2 Testování na FIRE databázi

Následně byly algoritmy otestovány na reálných snímcích. K testování byly využity snímky sítnice z databáze FIRE¹. Databáze FIRE obsahuje 134 párů snímků sítnice snímaných fundus kamerou Nidek AFC-210 s rozlišením 2912x2912. Snímky sítnic 39 mužů a žen ve věkovém rozmezí 19-67 let byly získány v roce 2006-2015. V závislosti na charakteru snímků sítnice jsou páry rozděleny do kategorií S, P a A. Kategorie S obsahuje 71 párů snímků, které vykazují překrytí větší než 75 %, kde nejsou přítomny anatomické rozdíly mezi snímky. Kategorie P obsahuje 49 párů snímků, kde rovněž nejsou přítomny anatomické rozdíly mezi snímky, překrytí snímků je ovšem menší než 75 %. Kategorie A obsahuje 14 párů snímků s překrytím větším než 75 %, které obsahují anatomické rozdíly mezi snímky v páru z důvodu progresu retinopatie. Kategorie S a P také obsahují patologické změny, které mohou ovlivnit strukturu sítnice, rozdíly mezi snímky v páru však díky neměnnosti se retinopatií nejsou detekovány. Databáze rovněž obsahuje kromě masky vymezující zorné pole také anotovaná data ve formě deseti korespondujících bodů ve snímcích každého páru, jak lze vidět na obrázku níže.

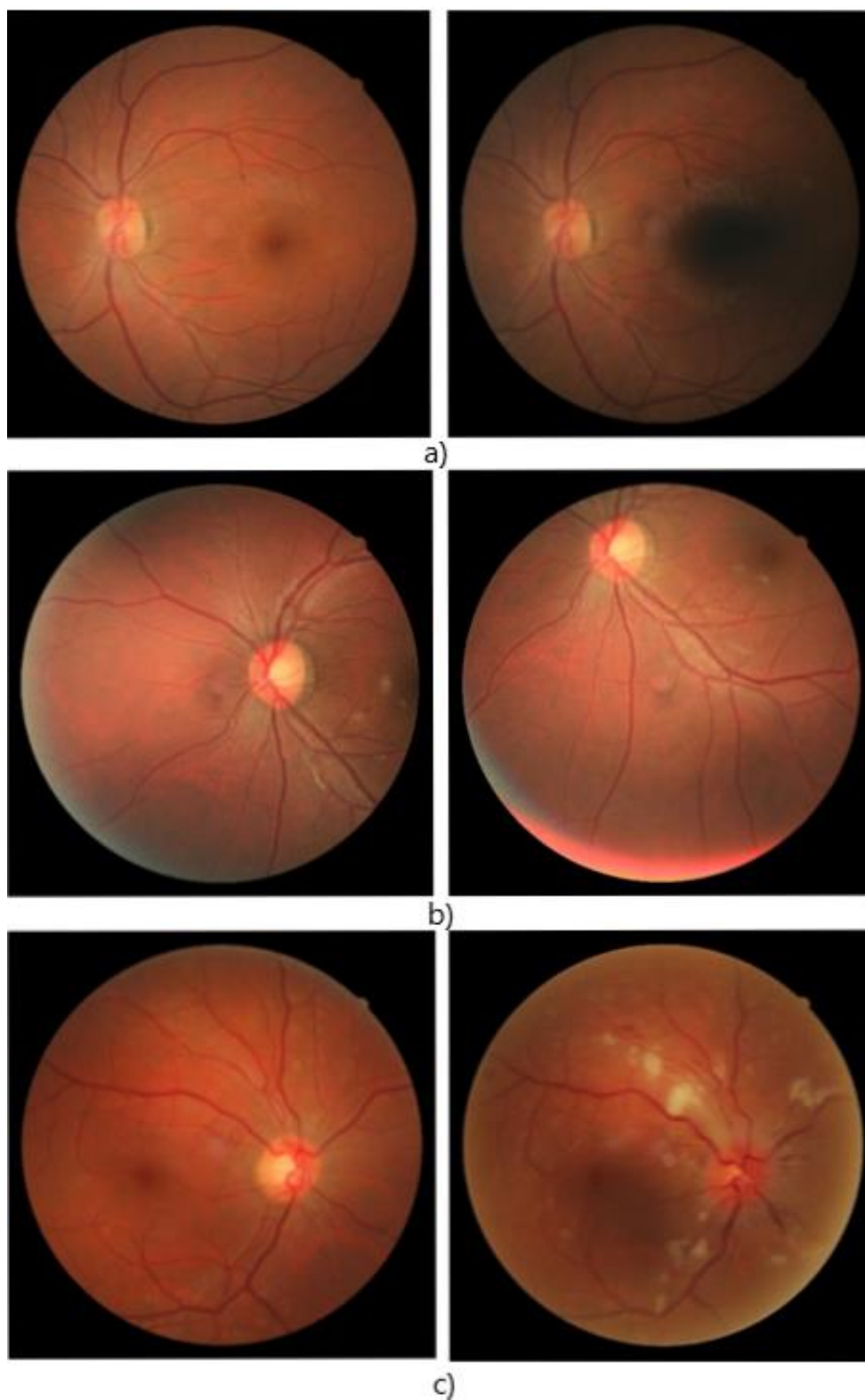


Obr. 16: Anotované body referenčního obrazu (vlevo) a pohybového obrazu (vpravo).

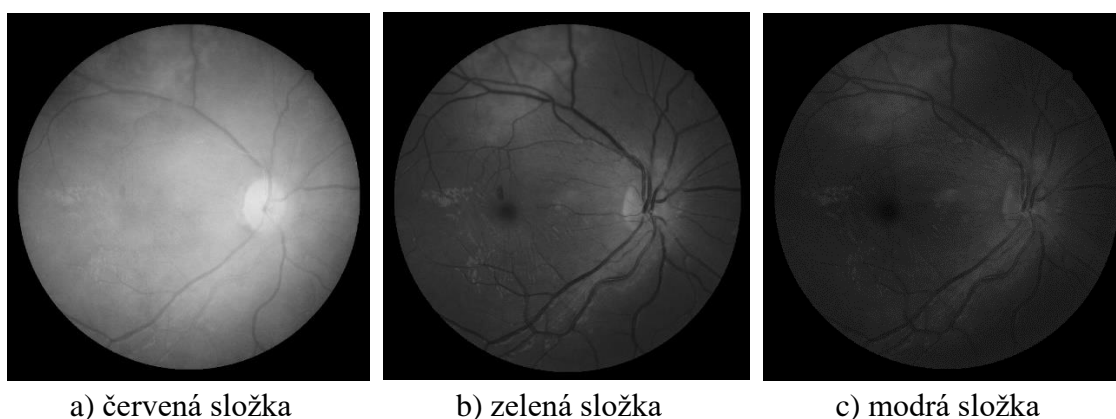
Na Obr 17. lze vidět příklady snímků spadajících do jednotlivých kategorií. Obr 17 a) obsahuje referenční a pohybový snímek kategorie S, konkrétně S37_01 a S37_02. Na části snímku lze vidět nerovnoměrné hodnoty jasu, což může být způsobeno makulární degenerací. Tyto rozdíly jasových hodnot mohou mít negativní vliv na kvalitu registrace. Na Obr. 17 b) jsou zobrazeny snímky patřící do kategorie P, konkrétně P13_01 a P13_02, jejichž překryv je menší než 75 % scény zorného pole. Obr. 17 c) obsahuje

¹ Veřejně dostupná na: <https://projects.ics.forth.gr/cvrl/fire/>.

snímky A05_01 a A05_02 spadající do kategorie A, kde lze vidět progres diabetické retinopatie, což může mít rovněž negativní vliv na kvalitu registrace.



Obr. 17: Příklady snímků jednotlivých kategorií.



Obr. 18: Rozdělení snímku A07_01 do jednotlivých barevných složek.

Při zobrazení snímku sítnice do jednotlivých barevných složek lze vidět, že v zelené složce jsou cévy sítnice a optický disk zastoupeny nejvýrazněji. Pro registraci bude proto využita pouze zelená složka. Originální snímky mají rozlišení 2912x2912. Pro urychlení algoritmu jsou tyto snímky podvzorkovány s faktorem 8. Nové rozlišení je tedy 364x364. Před registrací je vybrána oblast referenčního snímku, ze které bude počítána kritériální funkce. Tato oblast byla nejčastěji vybrána v okolí optického disku rozměru 100x100, jelikož zde dochází k výraznému větvení cév, což je pro porovnání obou obrazů žádoucí. Jako kritériální funkce byla zvolena střední kvadratická odchylka podobně jako v [48]. Je předpokládáno, že rozdíl mezi jednotlivými snímky bude nejznatelnější v translaci. Z principu snímání pomocí fundus kamery není rotace předpokládána ve velké míře, sejmutý obraz však může být proti referenčnímu obrazu mírně natočen. V této práci tak kromě rotace a translace v ose x a y další deformace předpokládány nejsou. Optimalizují se tedy tři parametry. Rotace je optimalizována v rozmezí $\langle -10; 10 \rangle$ stupňů a translace $\langle -200; 200 \rangle$ pixelů v obou osách. Ukončovacím kritériem všech evolučních algoritmů je proběhnutí maximálního počtu iterací.

3.2.1 Včelí algoritmus

Jak již bylo naznačeno v kapitole 3.1.1, funkci algoritmu ovlivňuje hned několik parametrů. Nejvýraznějšími parametry, které mají vliv na přesnost algoritmu je počet jedinců a počet iterací. Dalšími parametry jsou: počet nejlepších včel, počet elitních včel, počet vyčkávajících včel pro neelitní včely, počet vyčkávajících včel pro elitní včely a velikost prohledávaného lokálního okolí. Nalezení optimálních parametrů pro řešení určité úlohy nemusí znamenat optimální nastavení parametrů pro úlohu jinou. Vždy je potřeba nalézt optimální nastavení pro aktuální úlohu i dostupná data. Bylo vycházeno z parametrů použitých v několika publikacích, jako např [49] nebo [50]. Při testování algoritmu na snímcích sítnice však tato nastavení nevykazovala dobré výsledky.

Experimentálně tak bylo zjištěno nastavení parametrů, které vykazovaly lepší, nebo stejnou kvalitu v porovnání s ostatními nastaveními, viz Tabulka 1.

Tabulka 1: Počáteční nastavení parametrů včelího algoritmu.

<i>Parametr</i>	<i>Hodnota</i>
<i>Počet jedinců</i>	<i>300</i>
<i>Počet nejlepších včel m</i>	<i>13</i>
<i>Počet elitních včel e</i>	<i>2</i>
<i>Počet neelitních včel</i>	<i>m-e</i>
<i>Počet vyčkávajících včel vyslaných za neelitními včelami vn</i>	<i>10</i>
<i>Počet vyčkávajících včel vyslaných za elitními včelami</i>	<i>2vn</i>
<i>Počet iterací</i>	<i>40</i>

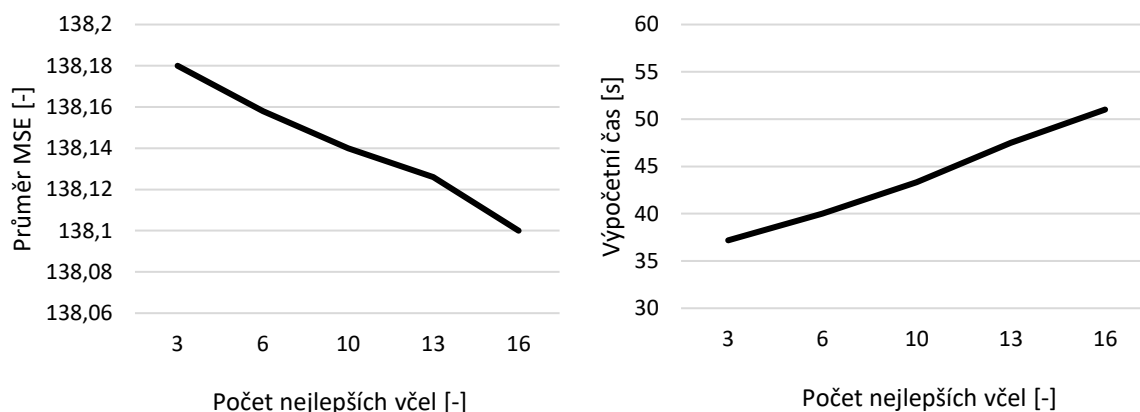
Z tohoto počátečního nastavení parametrů bude snahou získat parametry, při kterých program poskytuje co nejvyšší kvalitu registrace, ale zároveň registruje za co nejnižší čas. Je tedy nutné volit kompromis. Pro zjištění optimálních parametrů bylo vybráno deset snímků z každé kategorie databáze. Byly vybrány takové snímky, které nevykazovaly větší anatomické rozdíly mezi páry, kde nebylo ve větší míře přítomno nerovnoměrné osvětlení scény a které vykazovaly vysokou kvalitu registrace. Pro těchto 30 snímků byla jednotlivá nastavení parametrů spuštěna celkem desetkrát a bylo vypočítáno průměrné MSE, které bylo v závislosti na měnících se parametrech vyneseno do grafu. Kromě kritériální funkce byl analyzován také výpočetní čas. Testování probíhalo na stolním počítači, jehož parametry jsou uvedeny v Tabulce 2. Při testování jednotlivých parametrů bylo vycházeno z Tabulky 1 a byl vždy měněn pouze sledovaný parametr, ostatní parametry zůstaly nezměněny. Jelikož je kvalita registrace při počátečních podmínkách poměrně vysoká, jsou odchylky při změně jednotlivých parametrů ne příliš výrazné, a proto se špatné nastavení parametru vyznačí znatelnější změnou MSE v porovnání s ostatními nastaveními.

Tabulka 2: Parametry počítače.

<i>Procesor</i>	<i>Intel Core i7-4770 CPU</i>
<i>RAM</i>	<i>8 GB</i>
<i>Verze matlabu</i>	<i>Matlab R2020b</i>

3.2.1.1 Nejlepší včely

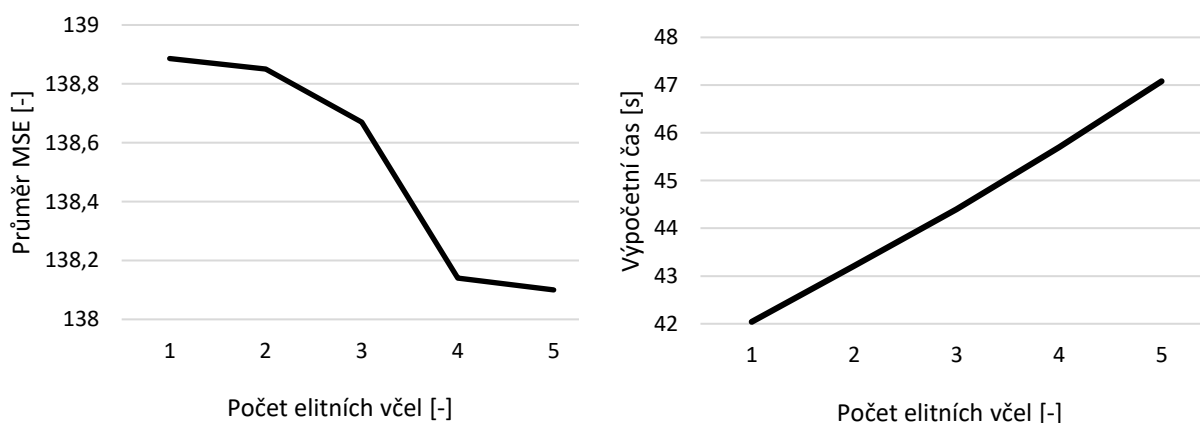
Nejlepší včely udávají nejkvalitnější včely, se kterými se v iteraci bude pracovat a budou se používat k postupnému zvýšení kvality. Závislost MSE na počtu nejlepších včel lze vidět na Obr. 19, kde s postupným zvyšováním počtu nejlepších včel mírně klesá MSE. Rovněž lze pozorovat změnu ve výpočetním času programu. Počáteční nastavení činilo deset nejlepších jedinců. Z grafu lze vidět, že rozdíl MSE mezi šesti a deseti jedinci není tak výrazný, naopak rozdíl výpočetní náročnosti činí skoro pět sekund.



Obr. 19: Závislost MSE a výpočetního času na počtu nejlepších včel.

3.2.1.2 Elitní včely

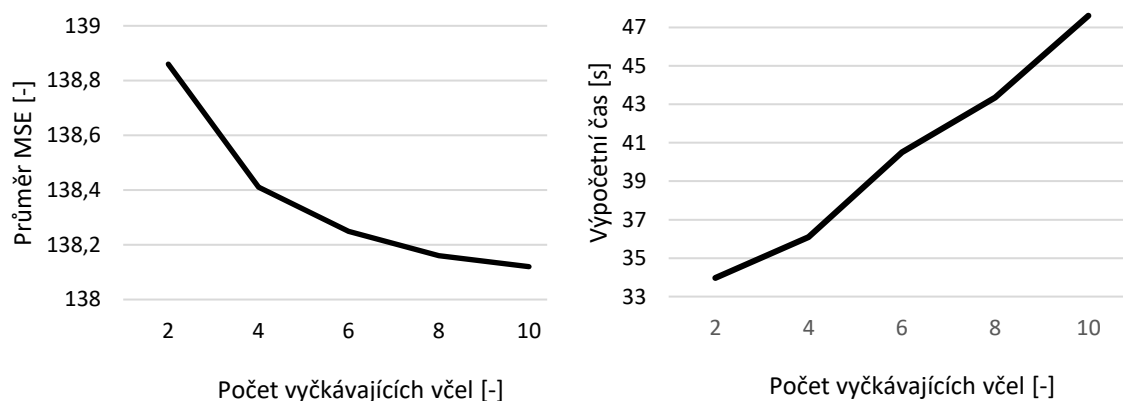
Elitní a neelitní včely udávají počet včel, které jsou vyslány za jednotlivými nejlepšími včelami. Za elitními včelami cestuje zpravidla více vyčkávajících včel než za neelitními. Závislost MSE a časové náročnosti na tomto parametru je vykreslena na Obr. 20. Z uvedených grafů si lze všimnout, že změnou z dvou elitních jedinců na tři by se zlepšila kvalita registrace za cenu mírně zvýšeného času, který činí skoro jednu sekundu.



Obr. 20: Závislost MSE a výpočetního času na počtu elitních včel.

3.2.1.3 Vyčkávací včely

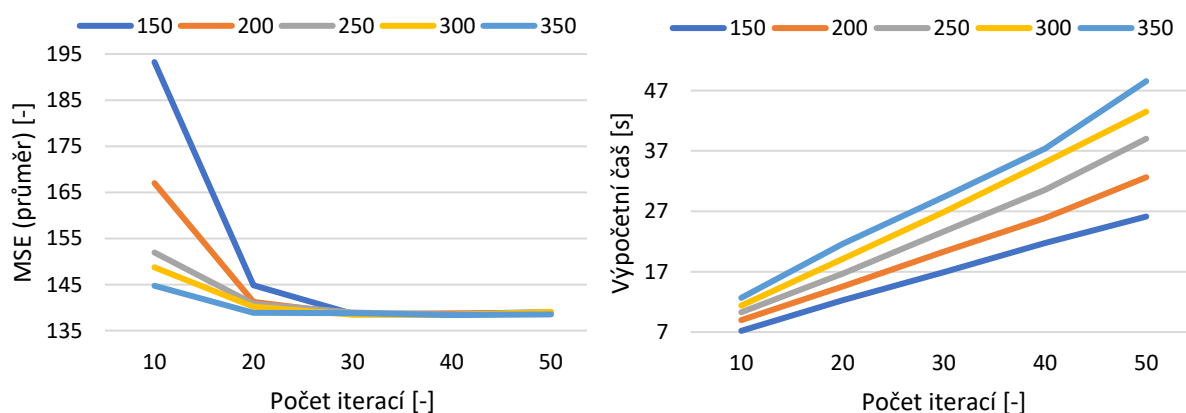
Při seřazení včel podle jejich kvality je za nejlepšími včelami vysláno několik vyčkávacích včel, které lokálně prohledávají prostor. Tento prostor je experimentálně stanoven na ± 2 od pozice aktuální včely. Po proběhnutí právě poloviny iterací je tento prostor zmenšen na jednu polovinu čili ± 1 . Lokální změny pozic včel jsou určeny náhodně. Rovněž platí, že za elitními včelami letí dvakrát více včel než za včelami neelitními, viz Tabulka 1. Jednotlivé závislosti na počtu vyčkávacích včel jsou uvedeny na Obr. 21, kde lze vidět, že změnou z deseti na osm vyčkávacích včel by bylo dosaženo víceméně stejné hodnoty MSE za cenu snížení výpočetní náročnosti o cca 4 sekundy.



Obr. 21: Závislost MSE a výpočetního času na počtu vyčkávacích včel.

3.2.1.4 Počet jedinců a iterací

Bezpochybně největší vliv na kvalitu registrace a výpočetní čas má nastavení parametrů počtu jedinců a celkový počet iterací. Je logické, že čím více je jedinců v populaci, tím kvalitnější bude registrace (ne vždy, roli v algoritmu hraje také náhoda), v takovém případě výrazně roste výpočetní čas algoritmu. Jednotlivé kombinace parametrů jsou znázorněny na následujících obrázcích.



Obr. 22: Závislost MSE a výpočetního času na počtu jedinců a iterací.

Na Obr. 22. si lze všimnout, že přibližně stejnou hodnotu naleznou všechna nastavení jedinců ve třiceti iteracích. Více iterací je tedy zbytečné uvažovat. Rovněž si lze všimnout, že přibližně stejných hodnot MSE dosahuje nastavení 350 jedinců a 20 iterací, 300 jedinců a 30 iterací, 250 jedinců a 30 iterací. Při pohledu na závislost výpočetního času lze konstatovat, že z těchto tří nastavení je výpočetně nejméně náročné nastavení s 20 jedinci a 350 iterací.

Při testování jednotlivých algoritmů bylo přistoupeno k některým změnám proti experimentálně určenému nastavení ve snaze zvýšit kvalitu registrace za cenu co nejméně narůstajícího výpočetního času. Jednotlivé parametry, které tvoří nutný kompromis mezi kvalitou registrace a výpočetní náročností jsou finálně shrnuty v Tabulce 3.

Tabulka 3: Finální nastavení parametrů včelího algoritmu pro registraci.

<i>Parametr</i>	<i>Hodnota</i>
<i>Počet jedinců</i>	<i>350</i>
<i>Počet nejlepších včel</i>	<i>10</i>
<i>Počet elitních včel e</i>	<i>3</i>
<i>Počet vyčkávacích včel vyslaných za neelitními včelami</i>	<i>8</i>
<i>Počet vyčkávacích včel vyslaných za elitními včelami</i>	<i>16</i>
<i>Počet iterací</i>	<i>20</i>

Při registraci párů snímků, které nebyly pro nastavování parametrů použity byla registrace v průměru mírně lepší pro výchozí nastavení, viz Tabulka 1. Díky změně parametrů se však výrazně snížila výpočetní náročnost, jež je nyní téměř dvakrát nižší. Výsledné hodnoty kritériální funkce a výpočetního času budou uvedeny v kapitole 3.2.4.

3.2.2 Genetický algoritmus

Podobně jako u včelího algoritmu nalezená nastavení parametrů pro genetický algoritmus, publikovaná v literatuře, nevykazovala skvělé výsledky. Proto bylo rovněž experimentálně stanoveno nastavení parametrů, při kterém program vykazoval relativně dobré výsledky, viz Tabulka 4. Pro optimalizaci parametrů byl zvolen podobný postup jako v případě včelího algoritmu, kdy byly měněny parametry: pravděpodobnost mutace, počet elitních jedinců, celkový počet jedinců a počet iterací.

Tabulka 4: Počáteční nastavení parametrů genetického algoritmu.

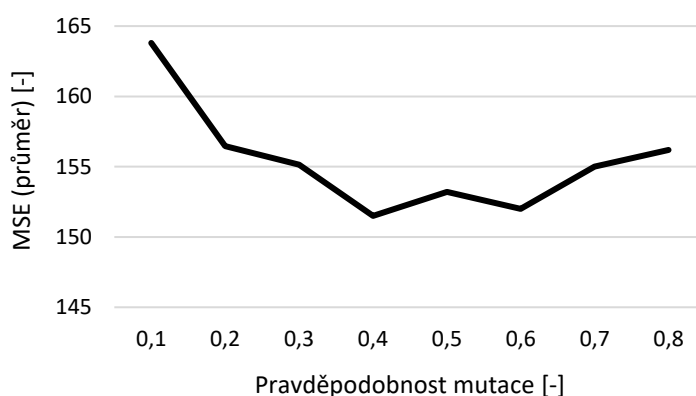
Parametr	Hodnota
Počet chromozomů	300
Pravděpodobnost mutace	0,4
Počet elitních chromozomů	4
Počet iterací	40

3.2.2.1 Pravděpodobnost mutace

Pravděpodobnost mutace udává pravděpodobnost, se kterou dojde k mutaci právě jednoho chromozomu, pro každý chromozom je tedy 40 % šance, že bude mutován. Mutace je v algoritmu reprezentována součtem aktuální hodnoty chromozomu a hodnoty definované následujícím vztahem:

$$10 \cdot \left(1 - \frac{it}{it_{max}}\right) \cdot (x - 0,5), \quad (34)$$

kde it je aktuální iterace, it_{max} je celkový počet iterací a x je náhodné číslo v rozsahu (0, 1). Z uvedeného vztahu plyne skutečnost, že hodnota mutace je v počátečních iteracích vyšší. Závislost MSE na pravděpodobnosti mutace lze vidět na Obr. 23. Nejmenší hodnoty MSE vykazuje algoritmus pro pravděpodobnost mutace 40, 50 a 60 procent. Změny výpočetního času v závislosti na pravděpodobnosti mutace jsou zanedbatelné, proto se výpočetní čas pro volbu parametru nebude uvažovat.



Obr. 23: Závislost MSE na pravděpodobnosti mutace.

3.2.2.2 Elitní jedinec

Elitní jedinec ovlivňuje rychlost konvergence k nalezenému nejlepšímu řešení v každé iteraci. Pokud je v první iteraci nalezeno řešení, které se blíží řešení optimálnímu, jednotliví jedinci se budou „stahovat“ k nalezenému řešení a křížením bude docházet

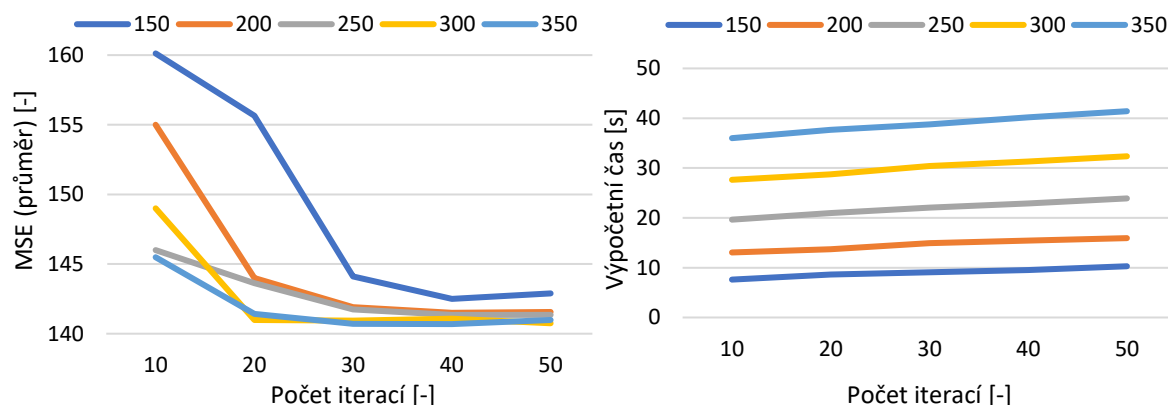
k usazování chromozomů kolem tohoto řešení, což je žádoucí. Jestliže se v prvních iteracích nepodaří nalézt řešení blízké optimu, bude docházet ke „stahování“ ostatních jedinců k elitním, kteří však nejsou přítomni blízko optimálního řešení, což má za následek konvergenci ke špatnému řešení, což je nežádoucí. Závislost MSE na počtu elitních jedinců lze vidět na Obr. 24. Výpočetní čas, stejně jako u pravděpodobnosti mutace, není při výběru optimálního počtu elitních jedinců uvažován, jelikož jsou změny zanedbatelné. Při pohledu na graf závislosti si lze všimnout, že při vypnutí elitismu (počet elitních jedinců je roven nule) vykazuje algoritmus nejhorší výsledky, postupným zvyšováním počtu elitních jedinců klesá hodnota MSE. Při počtu elitních jedinců větší než 10 je hodnota MSE srovnatelná, proto bude uvažováno deset elitních jedinců.



Obr. 24: Závislost MSE na počtu elitních jedinců.

3.2.2.3 Počet jedinců a iterací

Nejvýraznější vliv na hodnotu MSE a výpočetní čas má počet jedinců v populaci a celkový počet iterací. Vliv závislosti mezi jednotlivými parametry na hodnotu MSE a výpočetní čas je vykreslen na Obr. 25, kde si lze všimnout, že vyšší míru ovlivnění hodnoty MSE vykazuje počet jedinců nežli počet iterací. Relativně podobné výsledky vykazuje nastavení s 350 a 300 jedinci při dvaceti iteracích, dále se pak k podobné hodnotě MSE přibližuje nastavení s 250 jedinci při třiceti iteracích. Rovněž si lze všimnout, že výpočetní čas je taktéž více ovlivněn počtem jedinců, počet iterací ovlivňuje výpočetní čas jen mírně. Při těchto skutečnostech bude přistoupeno k nastavení 300 jedinců a 20 iterací. Výsledné parametry genetického algoritmu pro registraci lze vidět v Tabulce 5.



Obr. 25: Závislost MSE a výpočetního času na počtu jedinců a iterací.

Tabulka 5: Finální nastavení parametrů genetického algoritmu pro registraci.

Parametr	Hodnota
Počet chromozomů	300
Pravděpodobnost mutace	0,4
Počet elitních chromozomů	10
Počet iterací	20

3.2.3 PSO

Funkci PSO podobně jako u předchozích dvou algoritmů nejvíce ovlivňují parametry počet jedinců a počet iterací. Velmi důležitá je také správná souhra akceleračních koeficientů W , C_1 a C_2 . Pro nastavení akceleračních koeficientů bude vycházeno z [51] a jednotlivé parametry nastavení lze vidět v Tabulce 6.

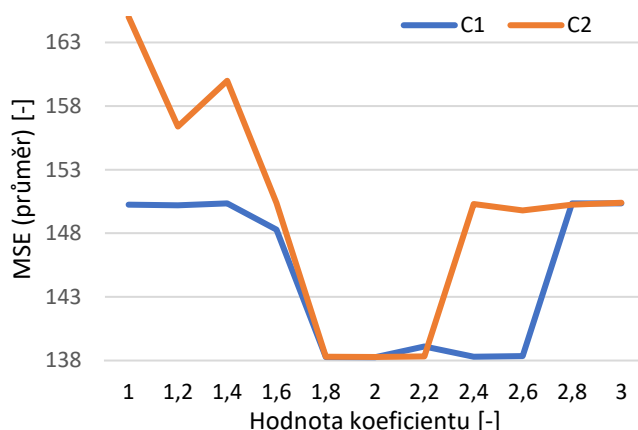
Tabulka 6: Počáteční nastavení parametrů PSO.

Parametr	Hodnota
Počet jedinců	300
C_1	2
C_2	2
W	1,8
Počet iterací	40

3.2.3.1 Akcelerační koeficienty C_1 , C_2

Koeficient C_1 při výpočtu aktuální rychlosti přikládá váhu nejlepší pozici individuálního jedince, při zvýšení tohoto koeficientu bude aktuální pozice více ovlivněna pozicí

v minulé iteraci. Naopak koeficient C_2 přikládá váhu nejlepší pozici v celé populaci, jeho zvýšením tak bude docházet k rychlejší konvergenci. Změny v nastavení těchto koeficientů výrazně ovlivňují hodnotu MSE a je nutné zkombinovat tyto parametry tak, aby algoritmus vykazoval co nejlepší výsledky. Změny těchto koeficientů nemají vliv na výpočetní náročnost programu. Vliv jednotlivých koeficientů lze vidět na Obr. 26.

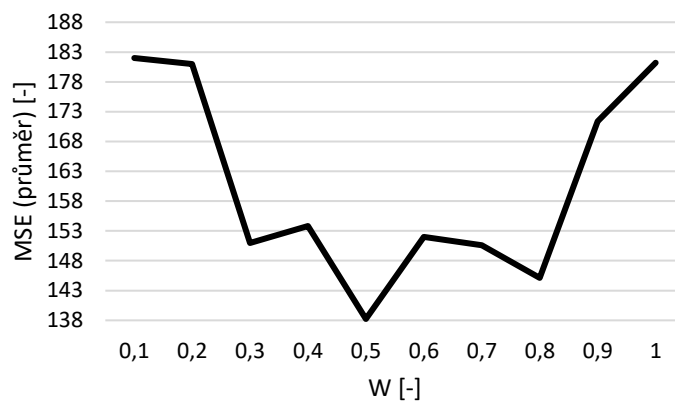


Obr. 26: Závislost MSE na parametrech C_1 a C_2 .

Z uvedených závislostí lze konstatovat, že větší vliv na hodnotu MSE vykazuje C_2 . Nejlepších výsledků vykazuje mezi hodnotami 1,8-2,2. Koeficient C_1 proti C_2 nevykazuje takové rozdíly v hodnotách MSE. Tyto rozdíly jsou ovšem stále znatelné. Nejlepší hodnoty C_1 jsou v rozmezí 1,8-2,6. S přihlédnutím na dříve odkazovanou publikaci [51] budou parametry, doporučené autory pro registraci, nezměněny.

3.2.3.2 Akcelerační koeficient W

Koeficient W udává vliv směru pohybu z předešlé iterace. Vliv koeficientu W si lze představit na následujícím příkladu. Dítě stojí na kopci a chystá se sjet kopec na saních. Po cestě na něho čeká kamarád, který si k němu má přisednout. Mohutně sněží a dítě na saních nevidí, kde přesně se kamarád nachází. Pozice kamaráda reprezentuje nejlepší pozici, která je předmětem optimalizace. Dítě na saních jede z kopce, po cestě však nikde kamaráda nenachází a nesnaží se brzdit. Následně si kamaráda všimne, sáně ale nabraly takovou rychlost, že bržděním sáně zastavují jen mírně. Toto je příklad hodnoty W , která je nastavena na vysokou hodnotu. Má výhodu ve smyslu zrychlení nalezení kamaráda, doladění konečné pozice však závisí na předchozí rychlosti. Pokud bychom nastavili W na nižší hodnotu, dítě na saních už od začátku bude předpokládat, že se v jeho blízkosti nachází kamarád, a proto bude více brzdit. Při nalezení kamaráda tak nemusí šlapat nahoru. To má však za následek zpomalení konvergence. Proto je nutné nalézt takovou hodnotu W , při které je řešení problému co nej kvalitnější.

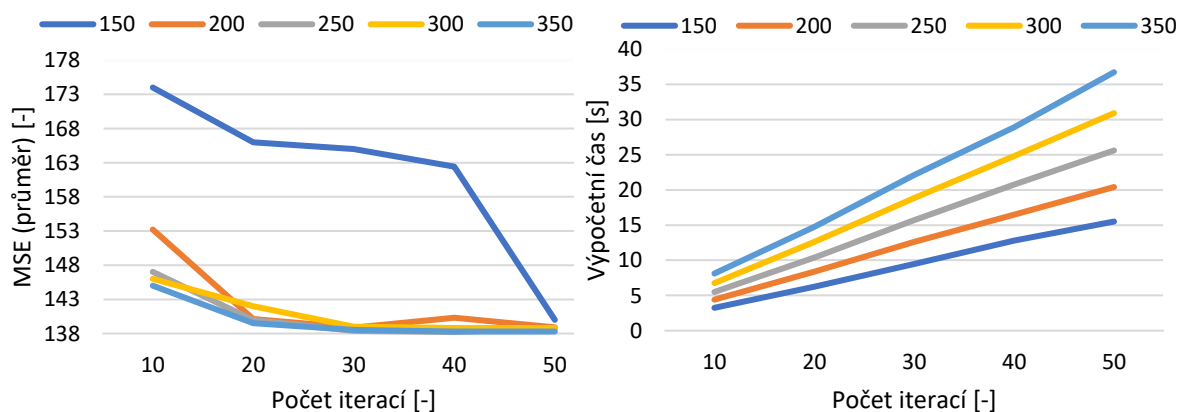


Obr. 27: Graf závislosti MSE na parametru W .

Na Obr. 27 lze vidět, že nejnižší hodnoty MSE vykazovaly hodnoty parametru mezi 0,3-0,8. Při testování se více osvědčila hodnota 0,5 proti hodnotě 0,8, která je doporučena autory v [51].

3.2.3.3 Počet jedinců a iterací

Také u tohoto algoritmu má největší vliv na MSE a výpočetní čas počet jedinců v populaci a celkový počet iterací. Jednotlivé závislosti lze vidět na Obr. 28. Z uvedených grafů lze konstatovat, že podobné hodnoty MSE vykazují jednotlivé počty jedinců do třiceti iterací, následný počet iterací již nemá vliv na snížení hodnoty MSE. To však neplatí pro nastavení se 150 jedinci, kdy při 50 iteracích dosahovala hodnota MSE v porovnání se 40 iteracemi výrazně nižších hodnot. Rovněž lze vidět, že výpočetní náročnost je více ovlivněna počtem iterací nežli počtem jedinců. Z tohoto důvodu bude zvolen nižší počet iterací s vyšším počtem jedinců. Výsledné nastavení jednotlivých parametrů lze vidět v Tabulce 7.



Obr. 28: Závislost MSE a výpočetního času na počtu jedinců a iterací.

Tabulka 7: Finální nastavení parametrů PSO pro registraci.

<i>Parametr</i>	<i>Hodnota</i>
<i>Počet jedinců</i>	300
C_1	2
C_2	2
W	0,5
<i>Počet iterací</i>	20

3.2.4 Výsledky registrace

Při nalezení jednotlivých parametrů translace a rotace byl na základě faktoru podvzorkování proveden přepočet jednotlivých parametrů odpovídajících nepodvzorkovanému obrazu. Po vynásobení nepodvzorkovaného pohybového obrazu s transformační maticí byly zjištěny mírné odchylky od referenčního obrazu. Proto bylo přistoupeno k optimalizaci parametrů pro nepodvzorkovaný obraz. Po ukončení algoritmu je vytvořena nová populace jedinců a k optimalizaci dochází opět. Rozmezí hodnot nové populace je dáno vztahem 35, 36 a 37. Zároveň je do nové populace přiřazen jedinec, který přesně odpovídá součinu nalezených hodnot a faktoru podvzorkování (pouze u translace, rotace zůstává nezměněna). Tímto dojde k doladění hledaných parametrů a přesnějšímu slícování obrazů.

$$\theta_n = \theta_p \pm 0.5, \quad (35)$$

$$x_n = x_p \cdot f \pm \frac{f}{2}, \quad (36)$$

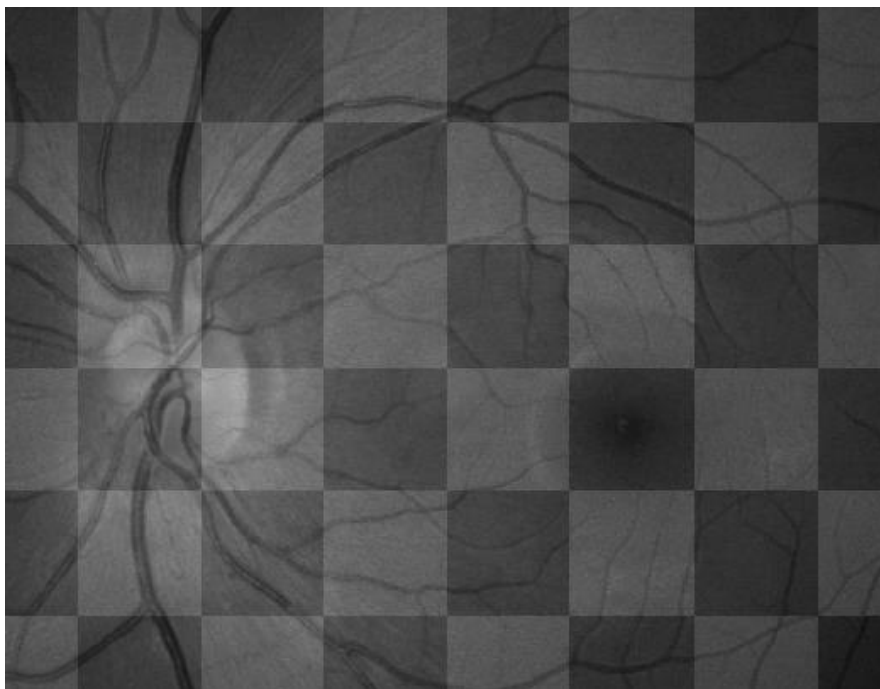
$$y_n = y_p \cdot f \pm \frac{f}{2}, \quad (37)$$

kde θ_n je vygenerovaná hodnota rotace, x_n je vygenerovaná hodnota translace v ose x, y_n je vygenerovaná hodnota v ose y, θ_p je nalezená hodnota rotace pro podvzorkovaný obraz, x_p je nalezená hodnota translace pro nepodvzorkovaný obraz v ose x, y_p je nalezená hodnota translace pro nepodvzorkovaný obraz v ose y a f je faktor podvzorkování. Kriteriační funkcí je opět MSE, tentokrát se však počítá ze zvětšené oblasti. Tato oblast je určena násobkem oblasti pro nepodvzorkované obrazy a faktorem podvzorkování. Tímto způsobem je kriteriační funkce počítána ze stejné scény obrazu (oblast je pouze zvětšena důsledkem větších rozměrů obrazu). Počet iterací je při optimalizaci snížen na pět a počet jedinců je snížen na 1/6 původního počtu. K této redukci bylo přistoupeno se záměrem příliš nezvýšit výpočetní čas programu.

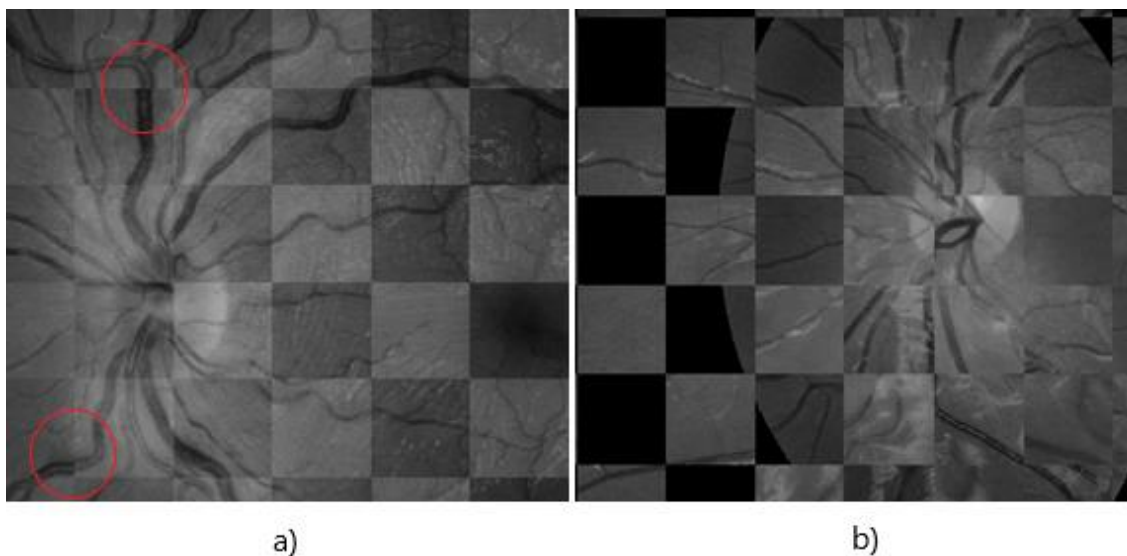
Pro otestování kvality registrace slouží deset anotovaných bodů, mezi kterými je počítána odchylka pixelů referenčního a pohybového obrazu. Odchylka je určena jako

vzdálenost mezi korespondujícími body referenčního a pohybového obrazu. Odchyšky v těchto bodech byly zprůměrovány a výstupem registrace jednoho páru snímků tak byla odchylka v ose x a odchylka v ose y. Tímto způsobem budou porovnány jednotlivé algoritmy. Pro porovnání bude sloužit také výpočetní čas, na který byl brán zřetel při nastavování jednotlivých parametrů.

Implementované algoritmy při registraci selhaly na několika párech snímků. Většinou byla nepřesná registrace způsobena nerovnoměrným osvětlením jednoho ze snímků nebo anatomickými rozdíly způsobující rozdíly jasových hodnot. Mírné odchylky u několika obrazů jsou také dány milným předpokladem, že pohybový snímek je od referenčního pouze posunut a rotován. Některé obrazy jsou zatíženy také jinými deformacemi, čímž může být ovlivněna přesnost lícování. Několik snímků bylo registrováno s relativně vysokou přesností nebo drobnými odchylkami. Pro vyšší přehlednost byly registrované snímky rozděleny do tří kategorií A, B, C. Do kategorie A spadají páry snímků, které vykazují relativně vysokou přesnost a odchylky nejsou ve velké míře zachyceny okem (průměrná odchylka menší než 10 pixelů). Kategorie B obsahuje obrazy, u kterých je registrace méně přesná a při porovnání obrazů lidským okem lze detekovat určité odchylky (průměrná odchylka menší než 20 pixelů). Do kategorie C spadají obrazy, u kterých byla registrace nepřesná (průměrná odchylka větší nebo stejná než 20 pixelů). Pro subjektivní porovnání jsou registrované snímky zobrazeny pomocí šachovnicového zobrazení. Příklady rozdělení do kategorií lze vidět na následujících obrázcích.



Obr. 29: Kategorie A – průměrné odchylky: $x=3,3$; $y=3,2$.



Obr. 30: a) kategorie B s průměrnými odchylkami: $x=4,3$; $y=33,5$, c) kategorie C s průměrnými odchylkami: $x=38$; $y=431$.

Na první pohled se registrované snímky na Obr. 30 a) mohou zdát velmi přesně slícované, při bližším prozkoumání však lze vidět určité odchylky, které jsou červeně znázorněny. Jelikož se v okolí těchto odchylek nacházejí anotované řídicí body, vedou tyto méně přesné registrované části ke zhoršení celkového průměru odchylky. Tabulka 8 obsahuje počty párů snímků rozřazené do jednotlivých kategorií.

Kategorie A – průměrná odchylka v ose x a y <10 pixelů.

Kategorie B – průměrná odchylka v ose x a y <20 pixelů.

Kategorie C – průměrná odchylka v ose x a y ≥ 20 pixelů.

Tabulka 8: Rozřazení lícovaných snímků udávající počet párů spadajících do jednotlivých kategorií. Kategorie S, P a A udávají přítomnost anatomických rozdílů a míru překryvu snímků, viz kapitola 3.2.

	Včelí alg.			Genetický alg.			PSO		
Kategorie	A	B	C	A	B	C	A	B	C
S	47	12	12	45	13	13	48	13	10
P	8	12	29	7	12	30	6	14	29
A	10	2	2	10	1	3	10	1	3

Tabulka 9: Srovnání algoritmů dle přesnosti a výpočetní náročnosti.

	Včelí alg.			Genetický alg.			PSO		
Průměr odchylky v obou směrech [px]	A	B	C	A	B	C	A	B	C
	4,81 ± 2,4	15,62 ± 4,12	328,4 ± 121,4	6,26 ± 3,6	17,26 ± 4,61	448,5 ± 214,9	6,87 ± 3,85	14,56 ± 4,98	412,8 ± 187,1
Průměr výpočetního času [s]	38,92			18,81			20,71		

Při pohledu na Tabulku 8 lze konstatovat, že implementované algoritmy nejlépe registrovaly páry, které jsou ve FIRE databázi přiřazeny do kategorie A a S. Odchylky registrovaných obrazů spadající do kategorie A byly nejčastěji způsobeny změnami v progresu retinopatie, což má za následek anatomické změny sítnice. Tyto anatomické změny potom znamenají rozdíly v jasových hodnotách referenčního a pohybového obrazu. Odchylky obrazů spadajících do kategorie S byly nejčastěji způsobeny nerovnoměrným osvětlením scény, některé části referenčního obrazu se tak jeví velmi tmavé nebo světlé proti korespondující části pohybového obrazu. Tyto jasové nerovnoměrnosti jsou v největší míře přítomny v obrazech spadajících do kategorie P, jejichž registrace byla z hlediska přesnosti lícování nedostačující. Při vyhodnocování přesnosti jednotlivých algoritmů nebude brána v úvahu kategorie C, jenž obsahuje páry snímků, které vykazovaly vysoké nepřesnosti a je prohlášeno, že algoritmy v registraci těchto snímků selhaly. Jestliže jsou vzaty v potaz pouze kategorie A a B, dosahuje v průměru nejvyšší přesnost včelí algoritmus, jenž má však proti zbylým dvěma algoritmům dvakrát vyšší hodnotu výpočetního času. Nižší přesnost registrace dosahuje genetický algoritmus a PSO, jejichž hodnoty jsou víceméně srovnatelné. Nepatrně lepší přesnost vykazuje PSO algoritmus, jehož výpočetní čas činí mírně vyšších hodnot než algoritmus genetický. Všechny evoluční algoritmy zvládly víceméně přesně registrovat snímky, jež neobsahovaly četnější anatomické rozdíly znamenající větší jasové rozdíly. Za tyto rozdíly může ve velké míře progres retinopatie nebo makulární degenerace. Snímky, které tyto jevy vykazovaly ve větší míře a nebylo je možné přesně registrovat, většinou spadají do kategorie C. Menší odchylky mohou být způsobeny milným předpokladem, že snímky jsou proti sobě pouze posunuty a rotovány. Zvýšení přesnosti registrace, zejména snímků, které obsahovaly jasové rozdíly díky anatomickým rozdílům, by bylo možné dosáhnout předzpracováním snímků různými formami filtrace nebo také zvolením kriteriální funkce, která není založena na porovnávání jasových hodnot obou snímků. Dále by se přesnost registrace dala zvýšit zvýšením počtu jedinců a iterací za cenu zvýšeného výpočetního času.

4. EVOLUČNÍ ALGORITMY V PŘÍZPŮSOBENÉ FILTRACI

Krokem předcházejícím segmentaci snímku může být pro zvýšení přesnosti přizpůsobená filtrace. Jak již bylo popsáno v kapitole 2.3.1, přizpůsobená filtrace využívá konvoluci masky s obrazem, čímž vzniká parametrický obraz, jehož odezva přímo závisí na míře korelace masky s daným obrazem. Tato maska je navržena na základě profilu objektu zájmu. Předmětem segmentace v této práci bude cévní řečiště snímků sítnice. Při navržení masky proto bude vycházeno z [52], kde masku autoři navrhovali dle profilu cévy sítnice. Přizpůsobená filtrace a následná segmentace bude prováděna na snímcích sítnice z veřejně dostupné databáze DRIVE². Databáze disponuje 40 snímky rozdělených do trénovacího a testovacího setu, každý obsahující dvacet snímků s rozlišením 565x584. Trénovací set obsahuje pro každý snímek zlatý standard. Testovací set obsahuje pro každý snímek dva zlaté standardy, manuálně označené pixely vyznačující cévní řečiště pomocí zkušených pozorovatelů. Pro každý snímek v databázi existuje také maska obrazu, která vymezuje zorné pole. Profil objektu zájmu – cévy – bude aproximován Gaussovou křivkou následujícím vztahem:

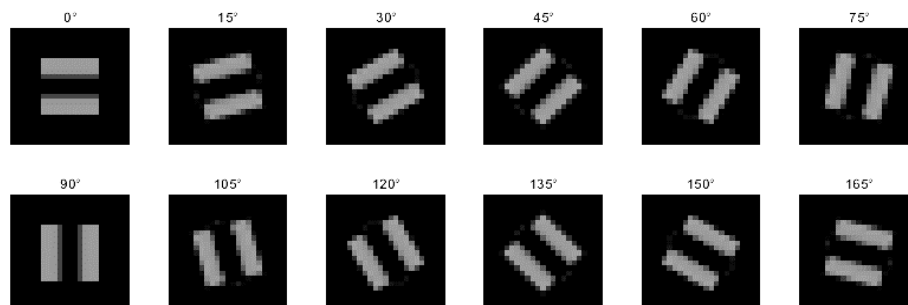
$$K(x) = -e^{\left(\frac{-x^2}{2\sigma}\right)}, \quad (38)$$

kde x je zvolený vektor o délce, která určuje počet vzorků impulsní charakteristiky vektoru K , σ reprezentuje šířku poloviny Gaussovy křivky právě v polovině její délky. Výstupní vektor K je následně roztažen ve směru osy y tak, aby výsledná maska byla čtvercového typu podobně jako v [53]. Takto získaná maska je následně rotována až o 165 stupňů s krokem 15 stupňů. Je tedy získáno 12 masek, stejně jako v [54], autoři však realizovali masky, jež odpovídají profilům cév rozdělených celkem do pěti tříd. Pro každou třídu tedy existuje jedna maska, jejíž rotováním je získáno celkem 60 masek. Cílem této práce bude optimalizovat velikost masky a parametr σ tak, aby za použití právě jedny masky (s rotovanými celkem 12), byla přizpůsobená filtrace a následná segmentace co nejúčinnější.

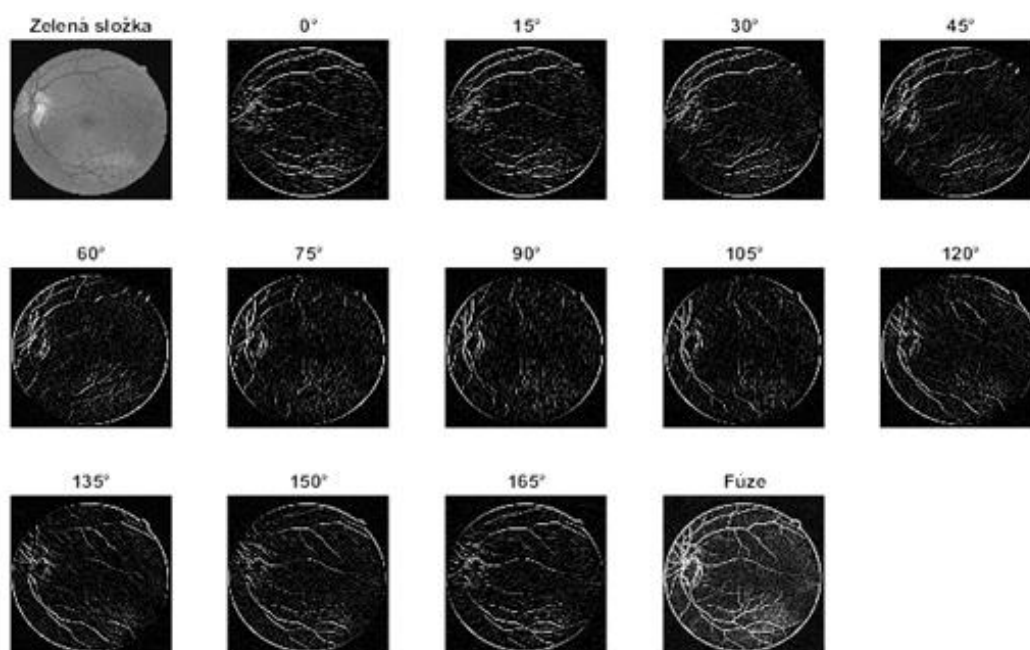
Pro nalezení optimálních parametrů (velikost masky, σ) bude využito trénovacího setu snímků databáze DRIVE. Každý snímek tohoto setu bude zpracováván jeden po druhém, čímž bude získáno dvacet řešení. Jednotlivé parametry budou následně zprůměrovány a použity pro segmentaci snímků v testovacím setu. Jak již bylo řečeno v kapitole 3.2, cévní řečiště je nejvýrazněji zastoupeno v zelené složce obrazu, proto bude výchozím snímkem při filtraci právě zelená složka snímku. Na základě optimalizovaných

² Veřejně dostupná na: <https://drive.grand-challenge.org/>

parametrů je sestrojeno 12 filtračních masek, jejichž konvolucí se zelenou složkou obrazu je získáno 12 parametrických obrazů, jejichž odezvy přímo závisí na korelaci masky s obrazem. S těmito 12 obrazy je provedena fúze na základě výběru maximálních hodnot (odezev) mezi korespondujícími pixely parametrických obrazů.



Obr. 31: Masky v jednotlivých směrech.



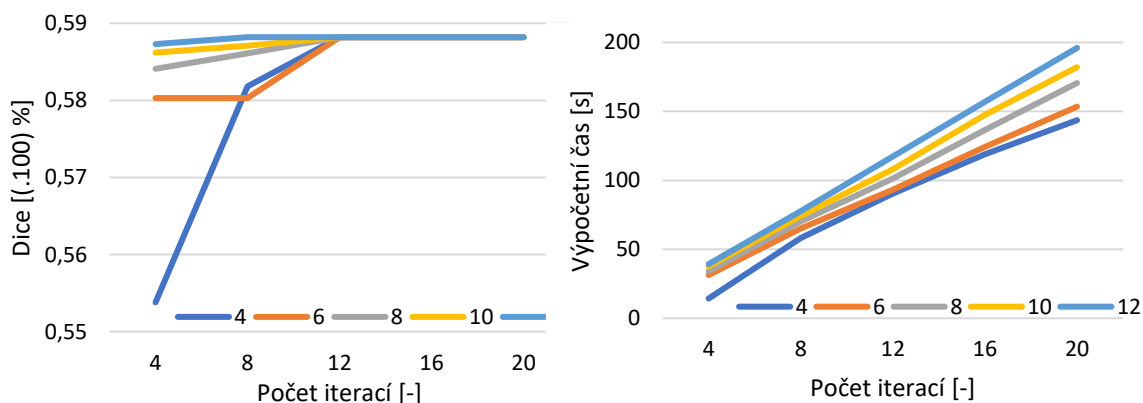
Obr. 32: Parametrické obrazy získané pomocí jednotlivých masek.

Pro tuto fúzi parametrických obrazů je následně zjištěn práh pomocí standardní Otsu metody za účelem binarizace cévního řečiště. Jako kritériální funkce slouží Sørensen-Dice koeficient mezi tímto binárním obrazem a zlatým standardem (anotovaná data) pro aktuální snímek sítnice. Čím vyšší je hodnota koeficientu, tím vyšší je podobnost mezi obrazy. Je tedy snaha o maximalizaci kritériální funkce. Je nutné zmínit, že dochází pouze k optimalizaci parametrů masky za účelem vytvoření parametrického obrazu. Nalezený práh pomocí Otsu metody slouží pouze k binarizaci snímku a jeho následnému porovnání s anotovanými daty za účelem zjištění hodnoty kritériální funkce. Na Obr. 31 lze vidět výsledek fúze parametrických obrazů, kde je přítomen i okraj zorného pole, který výrazně snižuje hodnotu kritériální funkce. Protože je však přítomný

ve všech fúzích parametrických obrazů, ovlivňuje hodnotu kritériální funkce u všech párů stejně a nemusí být odstraňován. Rozmezí hledaných hodnot pro velikost masky činí $\langle 5; 40 \rangle$ pixelů a pro parametr sigma $\langle 1; 40 \rangle$ pixelů. Hodnoty mohou nabývat pouze celých čísel. Experimentálně bylo zjištěno, že výrazně lepší výsledky dosahuje přizpůsobená filtrace při velikosti masky větší než sigma. Jestliže má jedinec, představující hledané parametry, hodnotu sigma vyšší, než je hodnota velikosti masky, přistupuje se k prohození těchto hodnot. Pokud tedy bude velikost masky = 10 px a sigma = 30 px, po prohození bude velikost masky = 30 px a sigma = 10 px. Pro výběr optimálních parametrů byly pozorovány různé kombinace parametrů v každém algoritmu. Pro porovnání proběhlo testování jednotlivých nastavení pro celou trénovací sadu a z maximální hodnoty kritériální funkce každého snímku byl vypočten průměr, na jehož základě probíhalo porovnání. Stejným způsobem byly pro jednotlivá nastavení získány hodnoty výpočetního času. Ukončovacím kritériem všech evolučních algoritmů bylo proběhnutí maximálního počtu iterací.

4.1 Včelí algoritmus

Jak již bylo uvedeno v kapitole 3.2.1, včelí algoritmus ovlivňuje několik parametrů. Těmi nejvýraznějšími je počet iterací a počet jedinců. Byly zkoumány různé kombinace počtu jedinců a iterací, a zjišťovány jejich kvality a výpočetní čas. Jednotlivé závislosti lze vidět na následujících obrázcích.



Obr. 33: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.

Na Obr. 33 lze vidět, že nejvyšší hodnotu podobnosti dosáhlo každé testované nastavení s minimálně dvanácti iteracemi. Také lze pozorovat, že i populace se čtyřmi jedinci při deseti a více iteracích dosahuje maximálních hodnot. Nejvyšších hodnot však vykazovalo nastavení s počtem 12 jedinců a 8 iterací, 10 jedinců a 12 iterací, 8 jedinců a 12 iterací. Při pohledu na závislost výpočetního času si lze všimnout, že k výraznějšímu ovlivnění dochází mnohem více ze strany počtu iterací nežli ze strany počtu jedinců. S ohledem na tyto závislosti bylo zvoleno 12 jedinců a 8 iterací. Závislosti zbývajících parametrů mají srovnatelný průběh jako závislosti uvedené v kapitole 3.2.1, díky kterým

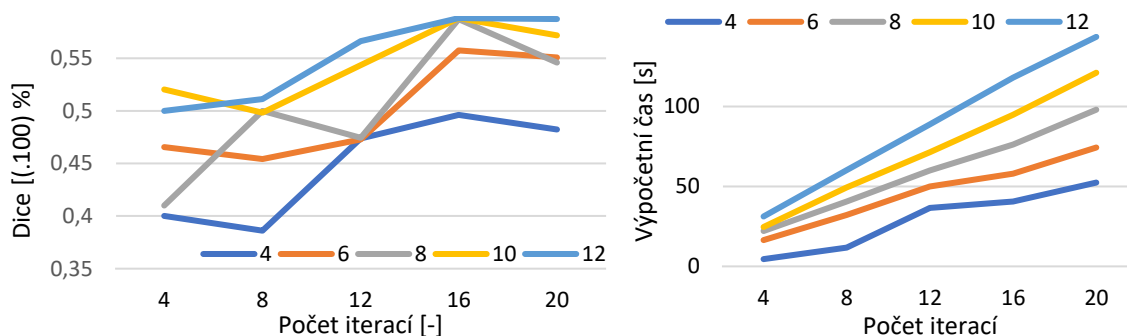
bylo přistoupeno k výslednému nastavení uvedeném v Tabulce 10. Rozsah lokálního prohledávání byl stanoven na ± 5 od aktuální hodnoty včely.

Tabulka 10: Finální nastavení včeliho algoritmu pro přizpůsobenou filtraci.

Parametr	Hodnota
Počet jedinců	12
Počet nejlepších včel	3
Počet elitních včel	2
Počet vyčkávajících včel vyslaných za neelitními včelami	2
Počet vyčkávajících včel vyslaných za elitními včelami	4
Počet iterací	8

4.2 Genetický algoritmus

Při pohledu na závislost koeficientu podobnosti na počtu iterací a jedinců lze konstatovat, že zásadnější roli hraje v hodnotách kritériální funkce počet jedinců nežli počet iterací. Na rozdíl od včeliho algoritmu, kde i nastavení se čtyřmi jedinci dosahovalo maximálních hodnot, je genetický algoritmus více závislý na počtu jedinců. Pravděpodobnost mutace a počet elitních jedinců má srovnatelnou závislost jako v kapitole 3.2.2 a s ohledem na závislost výpočetního času bylo dospěno k nastavení, jež je uvedeno v Tabulce 11. Rozsah mutace je podobně jako u registrace stanoven vztahem 34.



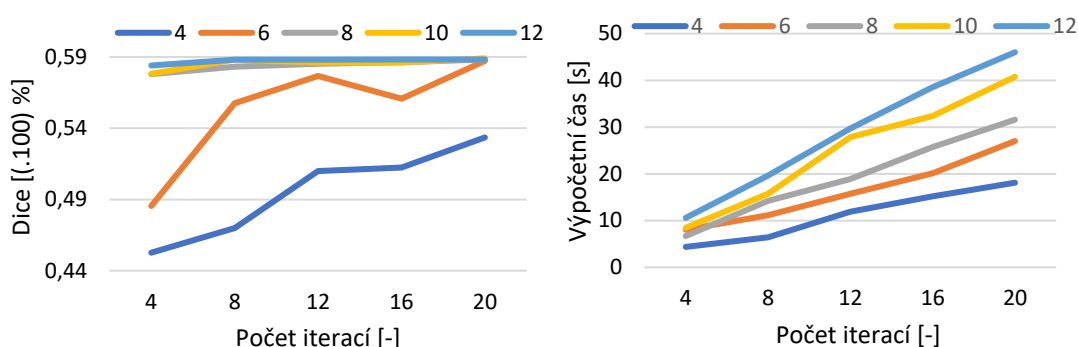
Obr. 34: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.

Tabulka 11: Finální nastavení genetického algoritmu pro přizpůsobenou filtraci.

Parametr	Hodnota
Počet chromozomů	12
Pravděpodobnost mutace	0,4
Počet elitních chromozomů	2
Počet iterací	10

4.3 PSO

Na závislosti hodnoty kritériální funkce na počtu jedinců a iterací lze vidět, že nejvyšší hodnoty koeficientu podobnosti vykazují vyšší počty jedinců již při nižším počtu iterací. Co se týče výpočetního času, tak je v porovnání s ostatními algoritmy velmi nízký. S ohledem na tuto skutečnost může být pro nastavení parametrů použit maximální počet jedinců i iterací. Při pohledu na průběh závislosti Dice koeficientu, kde nastavení s počtem jedinců 8, 10 a 12 vykazují podobné hodnoty, je takové nastavení zbytečné. Bude tak přistoupeno k nižšímu počtu iterací, čímž dojde k úspoře výpočetního času. Akcelerační koeficienty vykazují totožné závislosti jako v kapitole 3.2.3. Výsledné nastavení je uvedeno v Tabulce 12.



Obr. 35: Závislost Dice koeficientu a výpočetního času na počtu iterací a jedinců.

Tabulka 12: Finální nastavení parametrů PSO pro přizpůsobenou filtraci.

Parametr	Hodnota
Počet jedinců	12
C_1	2
C_2	2
W	0,5
Počet iterací	10

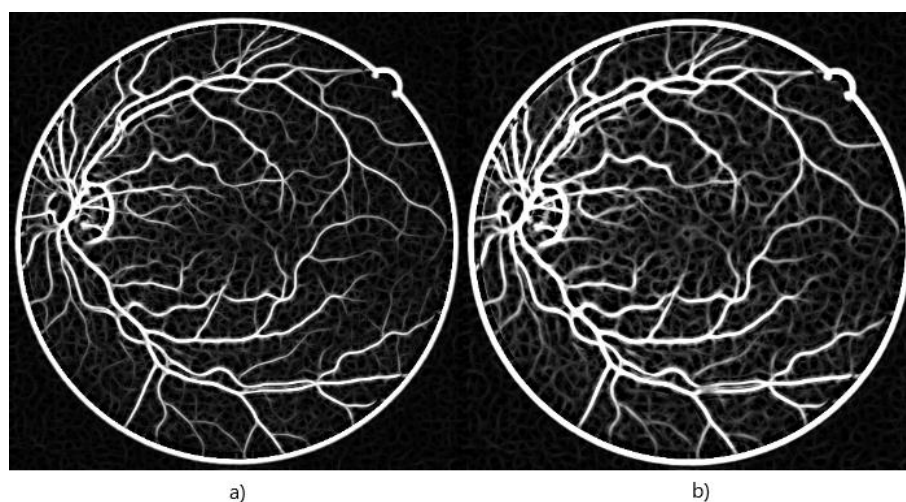
4.4 Srovnání jednotlivých algoritmů

Porovnání jednotlivých algoritmů bylo realizováno s využitím průměru maximálních hodnot koeficientu podobnosti Dice pro jednotlivé snímky z trénovacího setu DRIVE. Rovněž byl využit celkový výpočetní čas všech algoritmů. Jednotlivé hodnoty jsou uvedeny v Tabulce 13.

Tabulka 13: Parametry jednotlivých algoritmů.

	Včelí alg..	Genetický alg.	PSO
Dice [%]	59,24	56,23	58,44
Výpočetní čas [s]	1039,75	568,34	555,09
Nalezené parametry [px]	Velikost masky: 13 σ : 1	Velikost masky: 14 σ : 3	Velikost masky: 14 σ : 1

Nejlépe zvládl optimalizaci parametrů přizpůsobené filtrace včelí algoritmus, u kterého dosahovala průměrná hodnota Dice koeficientu nejvyšších hodnot. Tato skutečnost je však zastíněna velmi vysokým výpočetním časem, který je proti zbylým algoritmům téměř dvojnásobný. PSO vykazoval mírně nižší hodnoty podobnostního koeficientu, ze všech algoritmů má však nejnižší výpočetní čas. Genetický algoritmus vykazuje nejnižší hodnotu podobnostního koeficientu, hodnota výpočetního času je mírně vyšší než u PSO algoritmu. Při srovnání genetického algoritmu a PSO lze proto konstatovat, že lepších hodnot dosáhl algoritmus PSO. Optimální hodnota velikosti masky je 13 pixelů a hodnota sigmy činí 1 pixel. Jelikož byla hodnota sigmy hledána v rozsahu $<1, 40>$, odpovídá hodnotě na hranici rozmezí. Noví potomci genetického algoritmu mohou mít hodnoty pouze v rozmezí hodnot obou rodičů. Jestliže v počáteční populaci není vygenerován chromozom, který by vykazoval hodnotu sigma rovné jedné, není možné pouhým křížením dosáhnout požadované hodnoty. Jediným způsobem je potom dosažení hodnoty pomocí mutace, která může posunout hodnoty k nižším nebo vyšším hodnotám. Lze předpokládat, že pokud by optimální řešení odpovídalo prostředku hledaného rozsahu, genetický algoritmus by vykazoval vyšší kvalitu. Včelí algoritmus ve srovnání s PSO dosáhl vyšší hodnoty Dice koeficientu, ale vykazuje téměř dvojnásobnou hodnotu výpočetního času. Rozdíly fúzí parametrických obrazů získaných na základě včelího algoritmu a PSO jsou pouze mírné. Větší rozdíly lze vidět na obrazech získaných včelím algoritmem a genetickým algoritmem, viz Obr. 36.



Obr. 36: Fúze parametrických obrazů získaných na základě parametrů masky odvozené
a) včelím algoritmem, b) genetickým algoritmem.

Takto předzpracované obrazy jsou následně vstupem do samotné segmentace obrazu. V této práci bude přistoupeno k segmentaci pomocí prahování. Optimální práh bude určen metodou Otsu a bude předmětem optimalizace pro všechny algoritmy. Výpočet optimálního prahu bude určen z fúze parametrických obrazů získaných konvolucí zelené složky obrazu a masek, jejichž parametry odpovídají parametrům nalezených jednotlivými evolučními algoritmy. Optimalizace prahu pomocí PSO algoritmu bude tedy probíhat na fúzi parametrických obrazů získaných pomocí masky velikosti 14x14 pixelů a sigmoidu 2 pixely. Tato segmentace bude tentokrát probíhat na všech snímcích testovacího setu. Následně bude optimalizací hledána hodnota prahu, jež bude pro všechny obrazy v testovacím setu totožná. Při této optimalizaci budou pro všechny algoritmy použity stejné parametry masky pro přizpůsobenou filtraci. Tyto parametry budou odpovídat parametrům nalezených včelím algoritmem, jelikož vykazovaly nejvyšší hodnotu koeficientu Dice. Finální výstup segmentace bude k vidění v kapitole 5.4 na Obr. 41.

5. EVOLUČNÍ ALGORITMY V SEGMENTACI

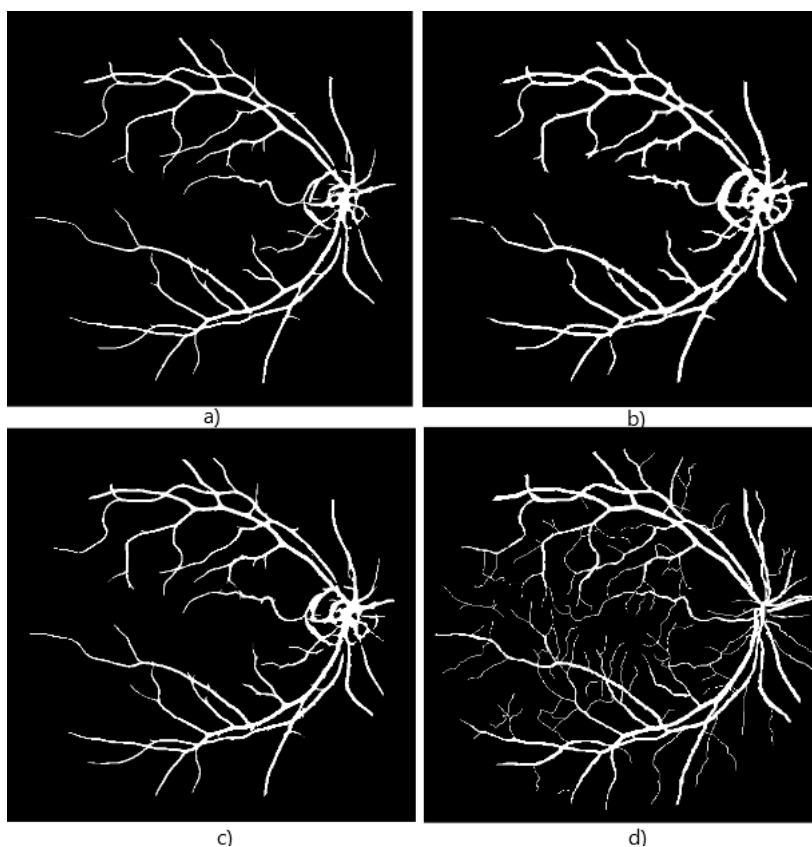
Jednou z nejčastěji používaných metod prahování je Otsu metoda, detailněji popsána v kapitole 2.2.1. Optimalizací bude hledán práh, jehož rozdělením do tříd bude získán maximální rozptyl tříd, který bude zároveň sloužit jako kritériální funkce. Bude tedy snaha o její maximalizaci. Hledaný práh je ovlivněn všemi jasovými hodnotami obrazu. Jelikož předmětem segmentace je cévní řečiště vyskytující se pouze v oblasti určené zorným polem, při stanovení prahu budou brány v potaz pouze hodnoty ležící uvnitř zorného pole. Hodnoty jedinců se pohybují v rozmezí $<0; 255>$ odpovídajícím možným jasovým hodnotám. Jedinci tedy budou představovat hodnoty prahu, které bude snaha optimalizovat takovým způsobem, aby hodnota kritériální funkce dosahovala maximální hodnoty. Hodnoty jedinců mohou nabývat pouze celých čísel.

Při testování se ukázalo, že pro nalezení ideálního prahu postačí jednotky jedinců i iterací. Jednotlivá nastavení proběhla na všech snímcích testovacího setu a byl stanoven jejich průměr. Výpočetní čas pro individuální nastavení parametrů se pohyboval v jednotkách setin sekundy, proto nebyl při výběru parametrů algoritmů uvažován. Závislosti počtu jedinců u jednotlivých evolučních algoritmů a počtu iterací s nastavením jednotlivých parametrů budou uvedeny v kapitole 5.1, 5.2 a 5.3. Algoritmy byly následně spuštěny na všech snímcích testovacího setu databáze DRIVE, kdy samotné optimalizaci prahu předcházelo předzpracování v podobně přizpůsobené filtraci s parametry nalezenými v kapitole 4. Parametry masky pro přizpůsobenou filtraci snímků v testovacím setu byly pro optimalizaci prahu včelím algoritmem tyto: velikost masky = 13 px a $\sigma = 1$ px, genetickým algoritmem: velikost masky = 14 px a $\sigma = 3$ px, algoritmem PSO: velikost masky = 14 px a $\sigma = 1$ px. Pomocí nalezeného prahu pro individuální snímek testovacího setu byl obraz segmentován. Ke snímkům testovacího setu je rovněž přiložen zlatý standard, na jehož základě dojde k vyhodnocení úspěšnosti segmentace dat. Bude stanovena senzitivita, specifita, celková přesnost a Dice koeficient pro každý snímek. Tyto parametry se ovšem nebudou počítat z celého obrazu, nýbrž pouze z oblasti odpovídající zornému poli. V oblasti za okrajem zorného pole totiž jasové hodnoty zlatého standardu i segmentovaného snímku odpovídají nule a výsledek by byl ovlivněn. V segmentovaném obrazu je rovněž přítomen zvýrazněný okraj zorného pole, který bude pro potřebu kvantitativního vyhodnocení odstraněn, jelikož by negativně ovlivňoval výsledek. Kvantitativní hodnocení bude provedeno pro každý snímek a výslednými parametry bude průměr těchto hodnot. Výsledky jsou uvedeny v Tabulce 14. Při porovnání algoritmů nebude hrát roli výpočetní čas, jelikož je zanedbatelný. Ukončovacím kritériem všech evolučních algoritmů bude proběhnutí maximálního počtu iterací.

Tabulka 14: Vyhodnocení úspěšnosti segmentace.

	Včelí alg.	Genetický alg.	PSO
Senzitivita	63,23 %	69,01 %	65,11 %
Specifická	98,07 %	95,12 %	97,10 %
Přesnost	93,62 %	91,78 %	93,01 %
Dice	71,36 %	68,03 %	70,15 %

Z uvedených hodnot v Tabulce 14 lze vidět, že nejlepší výsledky Dice koeficientu, přesnosti i specifity dosáhl včelí algoritmus, za kterým následuje PSO algoritmus, který ale vykazoval vyšší hodnotu senzitivity. Nejvyšší hodnotu senzitivity vykazoval genetický algoritmus, jehož další parametry však dosahovaly nejnižších hodnot. Senzitivita a specifita se většinou navzájem ovlivňují a zvýšením jednoho parametru se sníží parametr druhý, proto je nutné zvolit kompromis.



Obr. 37: Segmentovaný snímek 10.jpg na základě: a) včelího algoritmu, b) genetického algoritmu, c) PSO algoritmu. d) zlatý standard.

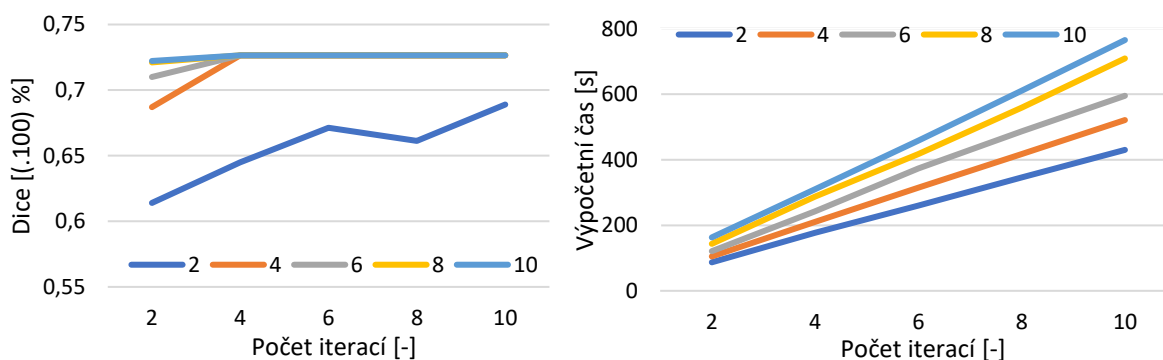
Uvedené výsledky jsou však velmi výrazně ovlivněny parametry masky, které byly použity pro přizpůsobenou filtraci. Při testování dvaceti snímků totiž každý parametr našel optimální hodnotu, která se lišila z důvodu různě filtrovaného snímku. Proto bude přistoupeno k hledání prahu ve snímcích filtrovaných pomocí masky, jejíž parametry vykazují pro všechny algoritmy stejné hodnoty.

Dalším předmětem optimalizace bude nalézt práh takové hodnoty, jenž bude shodný pro všechny snímky a současně bude v průměru vykazovat nejlepší výsledky segmentace pro celý testovací set. Kriteriační funkcí bude průměr Dice koeficientů všech segmentovaných snímků s aktuálním prahem. Opět tedy bude snaha o maximalizaci kriteriační funkce. Závislosti uvedené v následujících kapitolách jsou téměř totožné se závislostmi, které nebyly uvedeny pro algoritmy optimalizující práh pomocí OTSU, což je dáno hledáním optimálního parametru ve stejném rozmezí hodnot, tedy $\langle 0; 255 \rangle$. Jediným rozdílem je odlišná kriteriační funkce a zanedbatelný výpočetní čas, který se v případě OTSU metody pohyboval v desetinách sekundy z důvodu optimalizace prahu pouze pro jediný snímek. U optimalizace prahu, který bude společný pro všechny snímky je však výpočetní čas mnohem vyšší, jelikož je zapotřebí získat průměrného koeficientu Dice z dvaceti snímků. Parametry masky pro přizpůsobenou filtraci, která předchází optimalizaci prahu, budou tentokrát pro všechny algoritmy stejné, aby byla objektivně posouzena úspěšnost. Algoritmy by totiž vykazovaly jinou hodnotu Dice koeficientu i v případě nalezení optimálního prahu. Tyto rozdíly by byly způsobeny jinými parametry masky při přizpůsobené filtraci. Parametry masky budou odpovídat parametrům, které vykazovaly nejvyšší hodnotu Dice koeficientu. Takové parametry našel včelí algoritmus a odpovídají velikosti masky = 13 pixelů a $\sigma = 1$ pixel.

5.1 Včelí algoritmus

Největší vliv na hodnoty kriteriační funkce i výpočetního času má počet jedinců i iterací. Při pohledu na Obr 38. si lze všimnout, že algoritmus se dvěma jedinci vykazoval nejnižší hodnoty kriteriační funkce při všech počtech iterací. Nastavení se čtyřmi jedinci našlo optimální hodnotu ve čtyřech a více iteracích. Nejlepší výsledky vykazovala nastavení s osmi a deseti jedinci, u kterých hodnoty kriteriační funkce dosahovaly vysokých hodnot již při dvou iteracích a optimum našly ve čtyřech a více iteracích.

Výpočetní čas je více ovlivněn počtem iterací nežli počtem jedinců, jak lze konstatovat při pohledu na graf závislosti uvedený níže. Při výběru byl tedy zvolen vyšší počet jedinců s nižším počtem iterací.



Obr. 38: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.

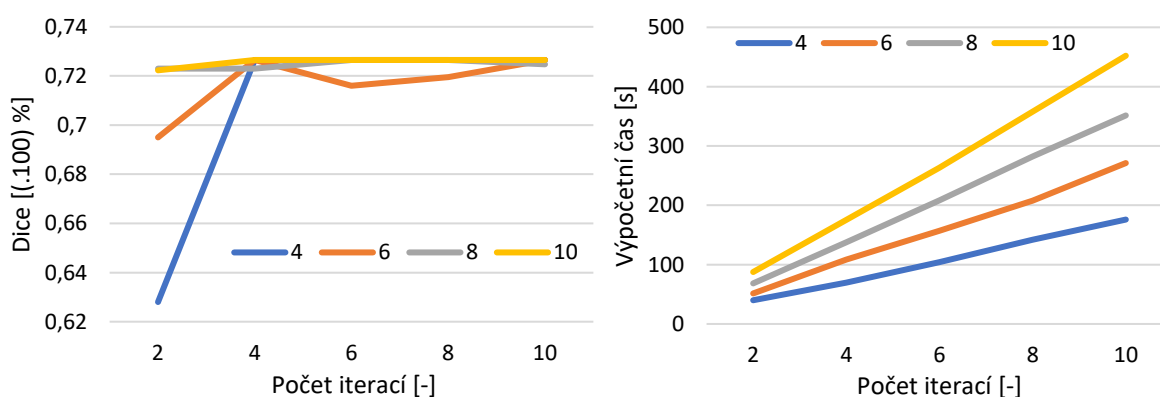
Zbylé parametry včelího algoritmu vykazovaly srovnatelný průběh jako závislosti uvedené v kapitole 3.2.1, proto nebudou dále uvedeny. Nastavení parametrů včelího algoritmu, které je totožné s nastavením pro optimalizaci prahu Otsu metodou je uvedeno v Tabulce 15. Rozmezí stanovené pro lokální prohledávání je totožné jako v případě přizpůsobené filtrace, tedy ± 5 .

Tabulka 15: Nastavení parametrů včelího algoritmu pro segmentaci.

Parametr	Hodnota
Počet jedinců	8
Počet nejlepších včel	2
Počet elitních včel	1
Počet vyčkávajících včel vyslaných za neelitními včelami	2
Počet vyčkávajících včel vyslaných za elitními včelami	4
Počet iterací	6

5.2 Genetický algoritmus

Závislosti genetického algoritmu byly testovány od čtyř do deseti jedinců s krokem dva. Počet jedinců roven dvěma byl vynechán. Jelikož je křížena lepší polovina jedinců, došlo by ke křížení jedince se stejným jedincem, což není možné. Nejvyšší hodnoty kritériální funkce vykazovala nastavení s šesti a osmi jedinci při více než čtyřech iteracích. Výpočetní čas dosahoval proti včelímu algoritmu nižších hodnot. Zbylé parametry genetického algoritmu vykazovaly srovnatelný průběh jako závislosti uvedené v kapitole 3.2.2, proto nebudou dále uvedeny. Nastavení genetického algoritmu, které je totožné s nastavením pro optimalizaci prahu Otsu metodou, je uvedeno v Tabulce 16. Rozsah mutace je podobně jako u registrace a přizpůsobené filtrace stanoven vztahem 34.



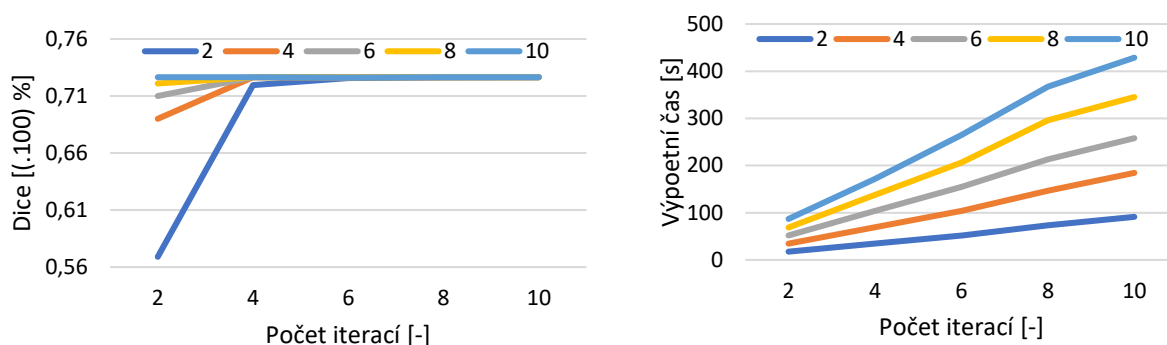
Obr. 39: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.

Tabulka 16: Nastavení parametrů genetického algoritmu pro segmentaci.

Parametr	Hodnota
Počet chromozomů	12
Pravděpodobnost mutace	0,4
Počet elitních chromozomů	2
Počet iterací	10

5.3 PSO

Stejně jako u genetického a včelího algoritmu vykazoval nejlepší hodnoty počet jedinců šest a více při počtu iterací vyšším než čtyři. Výpočetní čas dosahoval stejných hodnot jako u genetického algoritmu čili přibližně dvakrát nižší než u včelího algoritmu.



Obr. 40: Závislost Dice koeficientu a výpočetního času na počtu jedinců a iterací.

Akcelerační koeficienty C_1 , C_2 a W vykazovaly srovnatelný průběh jako závislosti uvedené v kapitole 3.2.3, proto nebudou dále uvedeny. Nastavení parametrů algoritmu PSO, které je totožné s nastavením pro optimalizaci prahu Otsu metodou je uvedeno v Tabulce 17.

Tabulka 17: Nastavení parametrů algoritmu PSO pro segmentaci.

Parametr	Hodnota
Počet jedinců	8
C_1	2
C_2	2
W	0,5
Počet iterací	6

5.4 Srovnání jednotlivých algoritmů

Před samotným vyhodnocením segmentace byly po zisku binárního obrazu uskutečněny dvě operace. První operací bylo zaplnění oblastí odpovídajících pozadí, které

neobsahovaly více než 50 pixelů. Druhou operací bylo odstranění oblastí odpovídajících popředí, jež obsahovaly méně než 100 pixelů. Takto upravený obraz byl následně podroben kvantitativnímu hodnocení. Pro porovnání jednotlivých algoritmů bude využit výpočetní čas nalezení prahu a podobnostní koeficient Dice, který udává procentuální podobnost segmentovaného obrazu a zlatého standardu. Dále senzitivita, která značí procento detekovaných cév algoritmem ze všech manuálně označených cév ve zlatém standardu. Také specifická, která značí procento správně detekovaných cév ze všech detekovaných cév. Často se udává také přesnost, jenž je poměrem součtu správně detekovaných cév a správně nedetekovaných cév se všemi pixely, ze kterých bylo prováděno hodnocení. Jednotlivé parametry byly získány z každého snímku testovacího setu a byl vypočten průměr. Kvantitativní hodnocení je uvedeno v Tabulce 18. Pro porovnání jsou v Tabulce 19 uvedeny výsledky segmentace ostatních autorů, kteří rovněž pracovali s DRIVE databází.

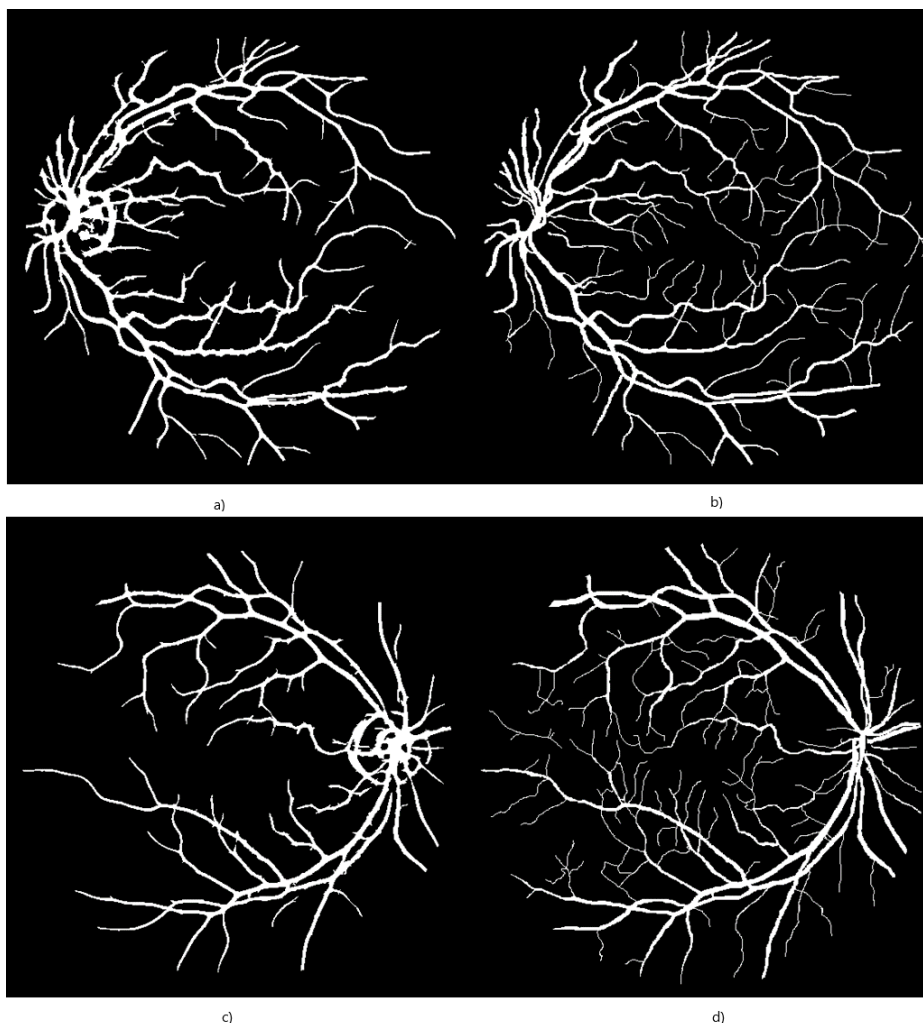
Tabulka 18: Kvantitativní hodnocení algoritmů a segmentace.

	Včelí alg.	Genetický alg.	PSO
Dice	72,83 %	72,58 %	72,83 %
Nalezený práh	91	98	91
Výpočetní čas	355,185 s	259,77 s	202,6 s
Senzitivita	72,17 %	69,54 %	72,17 %
Specifická	96,28 %	96,86 %	96,28 %
Přesnost	93,18 %	93,36 %	93,18 %

Tabulka 19: Kvantitativní hodnocení segmentace jiných autorů.

	Hassan a další [55]	Marin [56]	Vlachos a Dermatas [57]
Senzitivita	87,99 %	70,67 %	74,70 %
Specifická	97,99 %	98,01 %	98,00 %
Přesnost	96,25 %	94,52 %	96,00 %

V porovnání s uvedenými autory je úspěšnost segmentace v této práci spíše podprůměrná. Je ale nutné říct, že pro celý testovací set databáze DRIVE byly optimalizací zjištěny parametry masky přizpůsobené filtrace i prahu pro segmentaci, jež jsou stejné pro každý snímek testovacího setu. Včelí a PSO algoritmy našly totožný práh, jejich hodnoty jsou tedy identické kromě výpočetního času, který byl u PSO mnohem nižší. Lze tedy konstatovat, že vykazuje lepší výsledky. Genetický algoritmus vykazoval nižší hodnoty specifity, přesnosti i Dice koeficientu, zato vykazoval vyšší hodnoty senzitivity, podobně jako u optimalizace pomocí Otsu metody. Výpočetní čas genetického algoritmu dosahoval vyšších hodnot nežli u PSO algoritmu a nižších hodnot nežli u včelího algoritmu. Při porovnání Tabulky 18 a Tabulky 14 si lze všimnout, že Dice koeficient vykazoval vyšší hodnoty v případě použití jednoho prahu pro všechny snímky nežli výpočet prahu pro každý snímek zvlášť pomocí Otsu metody. To je dáno výběrem rozdílné kritériální funkce, kdy pro snímky, kde byl použit totožný práh, byl kritériální funkcí právě podobnostní koeficient Dice.

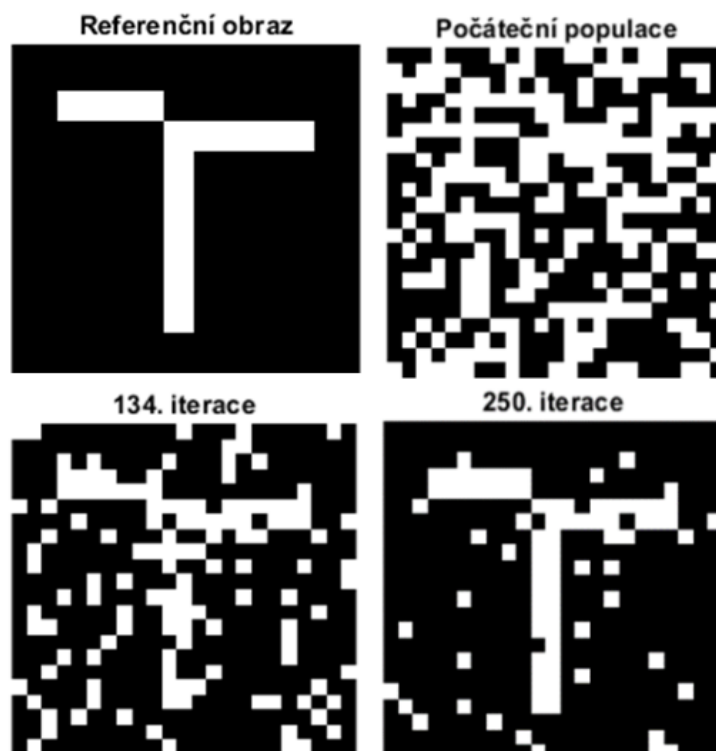


Obr. 41: a) segmentovaný snímek 01.jpg, b) jeho zlatý standard, c) segmentovaný snímek 10.jpg, d) jeho zlatý standard.

Pro zvýšení kvality segmentace by bylo vhodné zvolit vyšší počet masek pro přizpůsobenou filtraci podle profilu šířky cév podobně jako v [54], kde autoři zvolili pět masek. Jelikož je zvolena pouze jedna maska, nejsou detekovány užší a širší cévy současně. Kriteriační funkcí pro optimalizaci parametrů masky pro přizpůsobenou filtraci byl koeficient Dice, jenž značí podobnost mezi segmentovaným obrazem a zlatým standardem. Z principu je logické, že bude koeficient vykazovat vyšší hodnoty při zisku širších cév. Optimalizací jsou tedy zjištěny parametry masky, jejíž profil odpovídá širším cévám. Užší cévy nejsou segmentací zachyceny. Tato skutečnost znamená snížení senzitivity. Naopak specifitu může snižovat segmentovaný optický disk, který je detekován ve většině snímcích. Na Obr. 41 lze vidět dva segmentované snímky a jejich zlaté standardy. Na segmentovaných snímcích je absence užších cév a zvýrazněný optický disk, což má negativní vliv na úspěšnost segmentace snímků sítnice.

6. EVOLUČNÍ ALGORITMY V REKONSTRUKCI OBRAZU

Názorný příklad využití evolučních algoritmů uvedl autor v [58] na rekonstrukci obrazu, kde implementoval program na základě binárního genetického algoritmu, který dokáže rekonstruovat jakýkoliv binární obraz. Jako kritériální funkci zvolil Euklidovu vzdálenost mezi referenčním a rekonstruovaným obrazem. Na základě dvojbodového křížení a vážené rulety byli vytvářeni noví a kvalitnější jedinci, kteří měli větší podobnost s referenčním obrazem. Rekonstrukci binárního obrazu pomocí algoritmu, který byl z [58] implementován, lze vidět na Obr. 42. Binární genetický algoritmus je nejvhodnější metodou pro práci s binárními daty. Nevýhodou tohoto programu je však neschopnost rekonstruovat nebinární obrazy.



Obr. 42: Rekonstrukce binárního obrazu.

Jak již bylo zmíněno, tato rekonstrukce je možná pouze na binárních obrazech. Tato kapitola se bude zabývat evolučními algoritmy pro rekonstrukci šedotónových snímků.

6.1 Včelí algoritmus

Představme si referenční šedotónový obraz s rozměry 100x100. Nejprve je inicializována počáteční populace. Každý jedinec bude představovat obraz 100x100 s náhodně

generovanými jasovými hodnotami $\langle 0; 255 \rangle$. Mezi každým jedincem je spočítána Euklidova vzdálenost, což je kritériální funkce podobně jako v [55] a jedinci se seřadí podle kvality. Vybere se počet nejlepších včel, ke kterým jsou přiřčeny náhodné hodnoty podobně jako v dříve zmíněných aplikacích včelího algoritmu. Jeden jedinec je tedy představován 100x100 parametry (pixely). Tímto způsobem je prováděno lokální prohledávání. Některé jasové hodnoty se přiblíží korespondujícím pixelům v referenčním obrazu, ale některé zase vzdálí. Tímto způsobem, který je typický pro včelí algoritmus a funguje na optimalizaci jedinců s méně parametry, je téměř nemožné postupem iterací nalézt kvalitnější obraz, což bylo experimentálně ověřeno. Proto je nutné zvolit jiný přístup, který bude dále popsán.

Počátek algoritmu začíná vygenerováním počáteční populace s jedinci, kteří představují 100x100 obraz. Pro každého jedince je vypočtena Euklidova vzdálenost s referenčním obrazem a jsou seřazeni podle kvality. Podle této kvality je vybráno několik nejlepších včel pro další práci. Společně s Euklidovou vzdáleností mezi jedincem a referenčním obrazem je vypočtena i Euklidova vzdálenost každého pixelu (parametru) jedince s korespondujícím pixelem referenčního obrazu. Pracovat se tedy bude s Euklidovými vzdálenostmi celých obrazů (jedinců) a Euklidovými vzdálenostmi pixelů (parametrů jedince). Pro každého nejlepšího jedince je vytvořeno několik nových obrazů, jejichž jasové hodnoty pixelů se liší o hodnotu R od jasových hodnot korespondujících pixelů aktuálních nejlepších jedinců. Hodnota R je vyjádřena následujícím vztahem:

$$R = r \cdot (x - 0.5), \quad (39)$$

$$r = 100 \text{ pro } \frac{it_{max}}{it} > 3,5, \quad (40)$$

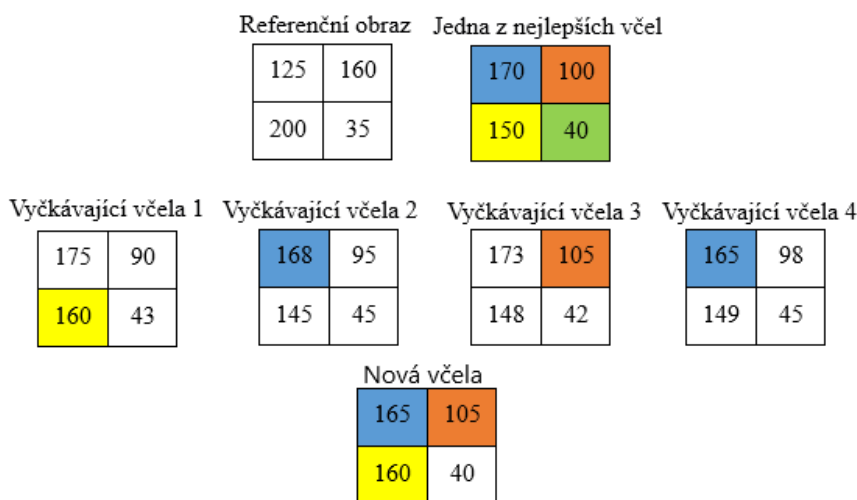
$$r = 50 \text{ pro } \frac{it_{max}}{it} > 2,5, \quad (41)$$

$$r = 25 \text{ pro } \frac{it_{max}}{it} \leq 2,5, \quad (42)$$

kde x je náhodné číslo v rozsahu $(0, 1)$, it_{max} je celkový počet iterací a it je aktuální počet iterací. Ze vztahů je tedy patrné, že s nabývajícím iteracemi rozsah lokálního prohledávání kolem aktuálního pixelu klesá.

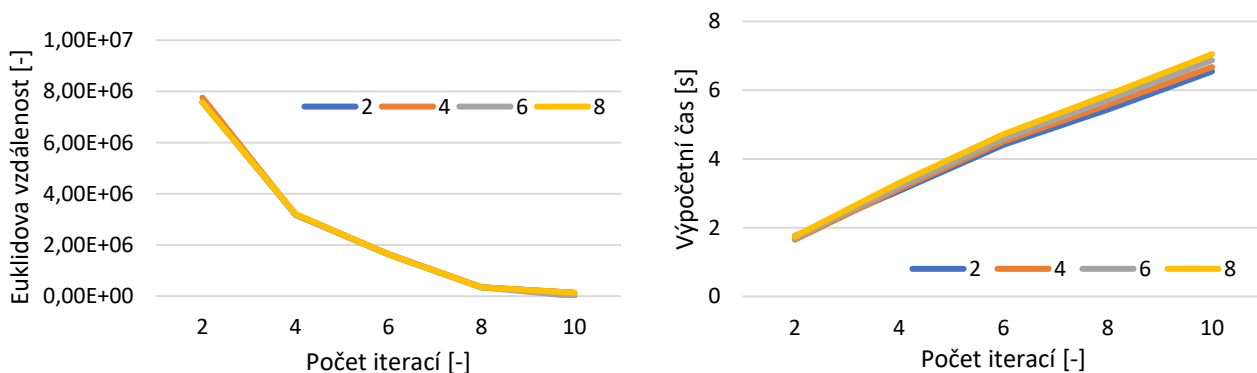
Počet nových obrazů je stanoven počtem vyčkávajících včel. Pro každý takto vytvořený obraz je vypočtena Euklidova vzdálenost každého pixelu s korespondujícím pixelem referenčního obrazu. V případě, že je počet vyčkávajících včel stanoven na čtyři, pak jsou pro pixel i, j každé z nejlepších včel vytvořeny čtyři nové hodnoty jasu (čtyři nové obrazy). Následuje porovnání, zdali má nějaký z nově vytvořených pixelů vyšší kvalitu než pixel, na základě kterého byly nové pixely vytvořeny. Jestliže nové pixely vykazují vyšší kvalitu, je starý pixel nahrazen pixelem nejvyšší kvality. Jestliže vyšší

kvalitu nevykazují, pixel zůstává nezměněn. Tento proces lze vidět na Obr. 43, kde byl pro ilustraci zvolen obraz s rozměry 2x2. Na obrázku jsou čtyři vyčkávající včely, jejichž pixely jsou porovnávány se včelou, za kterou přiletěly. Pixely vyčkávajících včel znázorněné modrou barvou vykazují vyšší kvalitu než pixel nejlepší včely, znázorněný také modrou barvou. Nové včele na stejné pozici bude tedy přidělena hodnota nejkvalitnějšího pixelu, v tomto případě pixelu náležící vyčkávající včele číslo čtyři. Stejným způsobem jsou včele přiděleny hodnoty ostatních pixelů.



Obr. 43: Vznik nových včel.

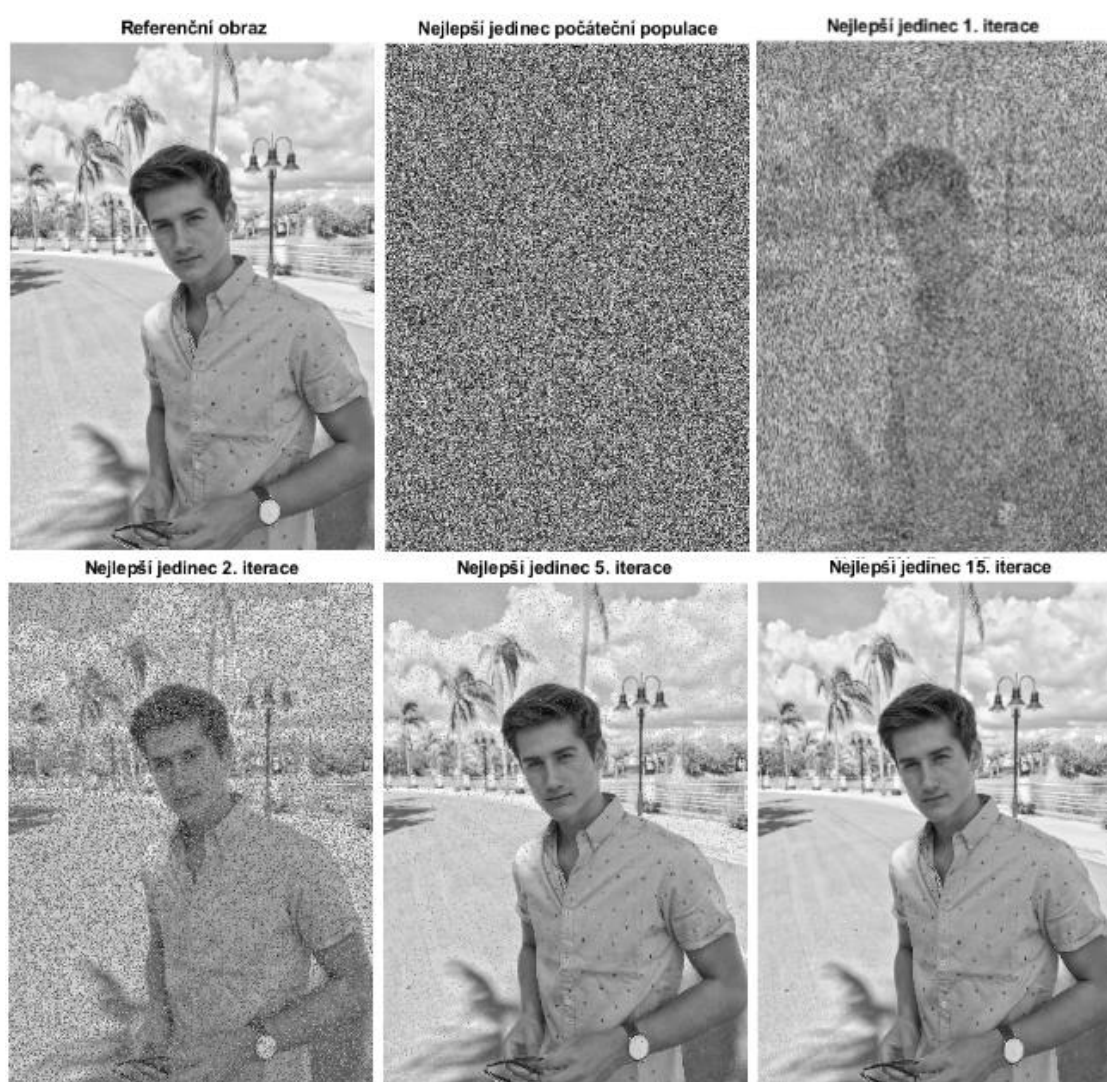
Takto je procházen celý obraz a méně kvalitní pixely jsou nahrazeny více kvalitními. Tento proces představuje lokální prohledávání. Získané obrazy představují novou populaci, jejíž doplnění do původního počtu jedinců je řešeno náhodným vygenerováním nových obrazů. Pro novou populaci obrazů je vypočtena Euklidova vzdálenost mezi jednotlivými obrazy a referenčním obrazem, je vybráno několik nejlepších obrazů a proces začíná znovu. Při testování se ukázalo, že hodnota kritériální funkce i výpočetní čas jsou mnohonásobně výrazněji ovlivněny počtem iterací nežli počtem jedinců. Ostatní parametry vykazují srovnatelnou závislost jako v předchozích kapitolách. Testování probíhalo na obrazu s rozměry 410x306.



Obr. 44: Závislosti kritériální funkce a času na počtu jedinců a iterací.

Tabulka 20: Nastavení parametrů včelího algoritmu pro rekonstrukci obrazu.

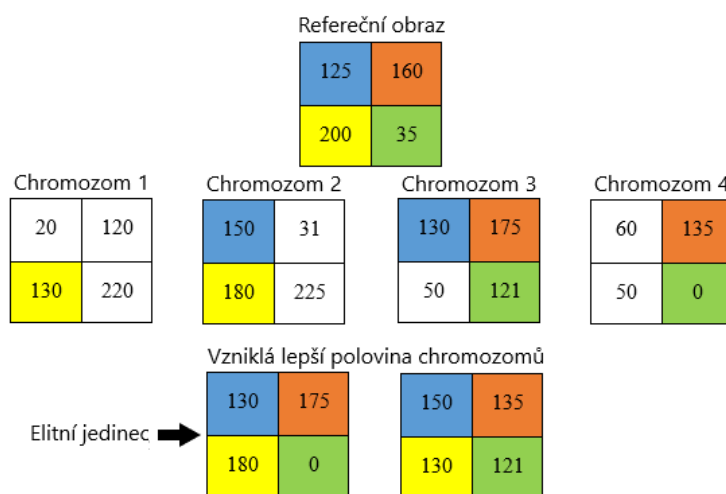
Parametr	Hodnota
Počet jedinců	4
Počet nejlepších včel	2
Počet elitních včel	1
Počet vyčkávajících včel vyslaných za neelitními včelami	4
Počet vyčkávajících včel vyslaných za elitními včelami	8
Počet iterací	15



Obr. 45: Proces rekonstrukce obrazu včelím algoritmem.

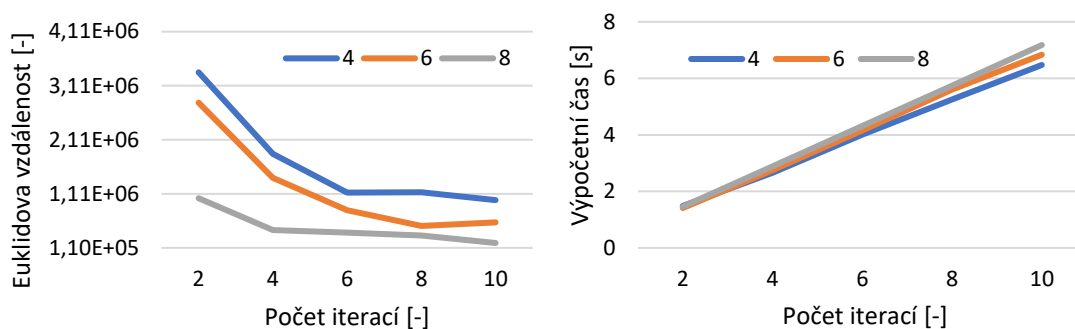
6.2 Genetický algoritmus

Podobně jako u včelího algoritmu musí být genetický algoritmus modifikován. Je vytvořena počáteční populace chromozomů představující jednotlivé náhodně generované obrazy. V každém obrazu je vypočtena Euklidova vzdálenost jasových hodnot všech pixelů s korespondujícími jasovými hodnotami pixelů referenčního obrazu. Následně je vybrána lepší polovina pixelů ze všech jedinců na dané pozici, čímž jsou vytvořeni jedinci podléhající reprodukci. Vznik lepší poloviny jedinců je ilustrován na Obr. 46, kde je situace popsána na obrazu s rozměry 2x2 a populaci se čtyřmi jedinci. Na obrázku lze vidět referenční obraz a jasové hodnoty všech pixelů. Také je zobrazena počáteční populace chromozomů rovněž s jasovými hodnotami pixelů. Nejprve je testován pixel referenčního obrazu, který je znázorněn modrou barvou. Je vypočtena Euklidova vzdálenost tohoto pixelu a pixelů náležícím jednotlivým jedincům a je vybrána lepší polovina. Tyto lepší pixely jsou znázorněny rovněž modrou barvou. Totožně jsou testovány všechny pixely obrazu. Následně vznikají nové chromozomy, a to takovým způsobem, že nej kvalitnější pixely jsou zkombinovány, čímž vzniká elitní jedinec. Dalším jedincem bude kombinace druhých nejlepších pixelů apod. Výše popsaný proces lze definovat jako „předkřížení“ chromozomů.



Obr. 46: Vznik lepší poloviny chromozomů.

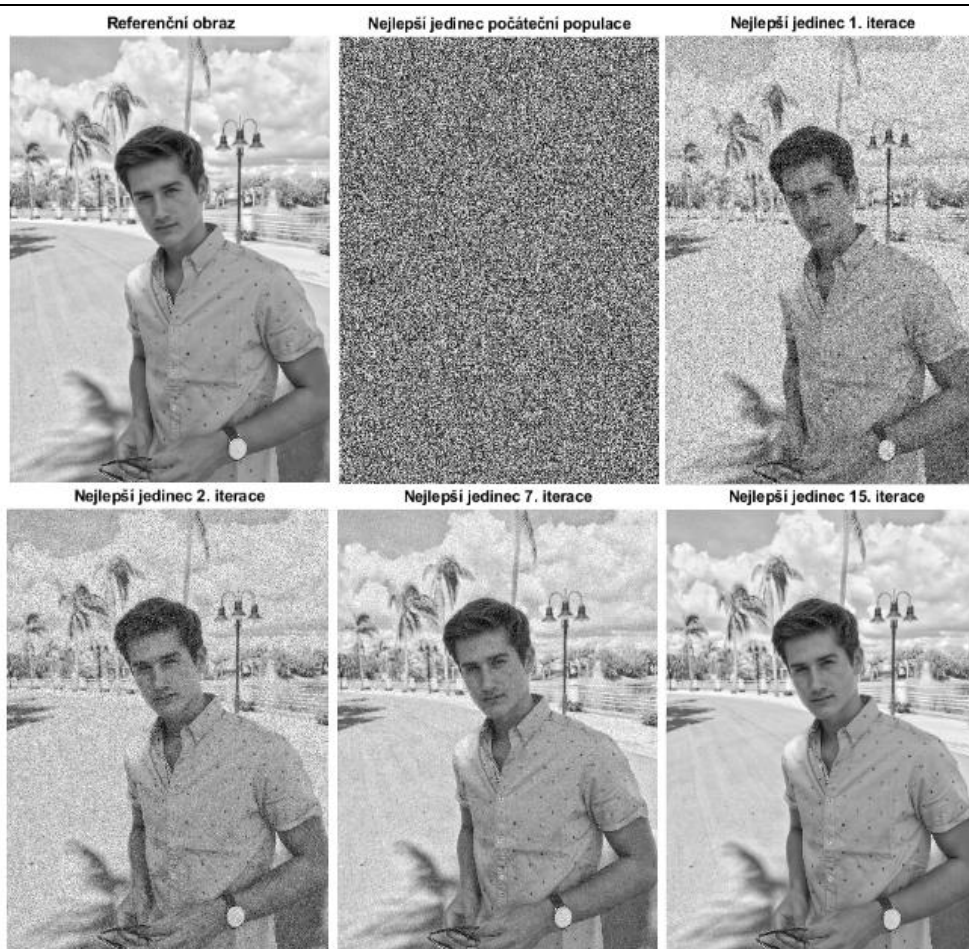
Vzniklá lepší polovina jedinců se následně podrobuje křížení, které probíhá standardním způsobem. Nově vzniklí potomci jsou následně mutováni přičtením náhodné hodnoty k jasové hodnotě pixelu s určitou pravděpodobností stejně, jak je popsáno v kapitole 3.2.2.1 vztahem 34. Jediným rozdílem je desetkrát vyšší hodnota mutace. Takto křížené a mutované chromozomy jsou zároveň novou populací pro další iteraci a proces se opakuje. Závislosti kritériální funkce a výpočetního času na počtu jedinců a iterací lze vidět na Obr. 47.



Obr. 47: Závislosti kritériální funkce a času na počtu jedinců a iterací.

Tabulka 21: Nastavení parametrů genetického algoritmu pro rekonstrukci obrazu.

Parametr	Hodnota
Počet chromozomů	10
Pravděpodobnost mutace	0,4
Počet elitních chromozomů	2
Počet iterací	15



Obr. 48: Proces rekonstrukce obrazu genetickým algoritmem.

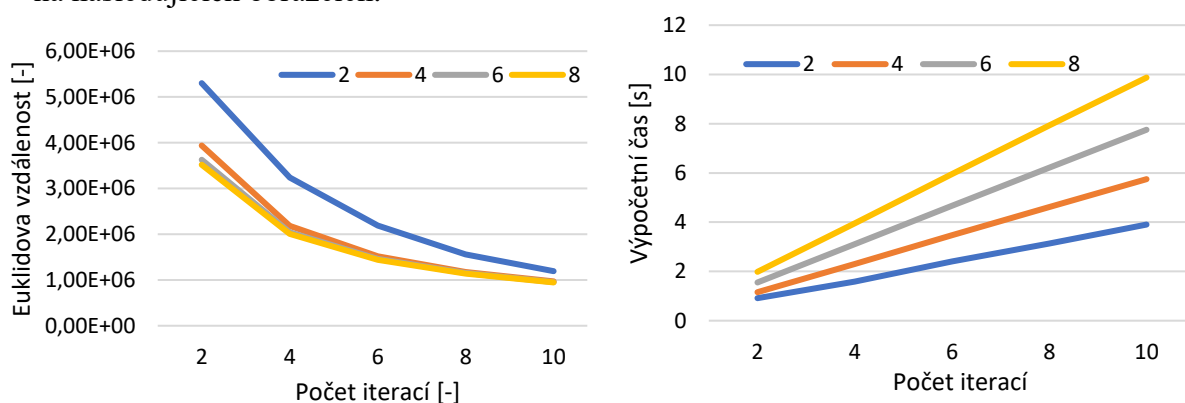
6.3 PSO

Stejně jako u předchozích zmíněných algoritmů musel být algoritmus PSO trochu modifikován. Počátkem algoritmu je inicializace počáteční populace, což představuje náhodně vygenerované obrazy. V každém obraze je určena kvalita všech pixelů a jsou určeny nejlepší pozice jasové hodnoty ze všech obrazů na jednotlivých pixelech podobně, jak je znázorněno na Obr. 49. Tyto nejlepší jasové hodnoty jsou uloženy jako nejlepší pozice celé populace a budou se využívat při přepočtu aktuální pozice a rychlosti. V každé iteraci jsou tedy všechny pixely a jejich rychlosti přepočítávány na základě jejich dosavadní nejlepší pozice a nejlepší pozice celé populace, jak je popsáno v kapitole 1.6.2.1 vztahem 4 a 5. Jediným rozdílem je, že pro populaci existuje tolik nejlepších pozic, kolik je pixelů v obraze a všechny pixely a jejich rychlosti každého jedince jsou aktualizovány zvlášť.

Referenční obraz							
125	160						
200	35						
Jedinec 1		Jedinec 2		Jedinec 3		Jedinec 4	
20	120	150	31	130	175	60	135
130	220	180	225	50	121	50	0

Obr. 49: Stanovení nejlepších pozic každého pixelu.

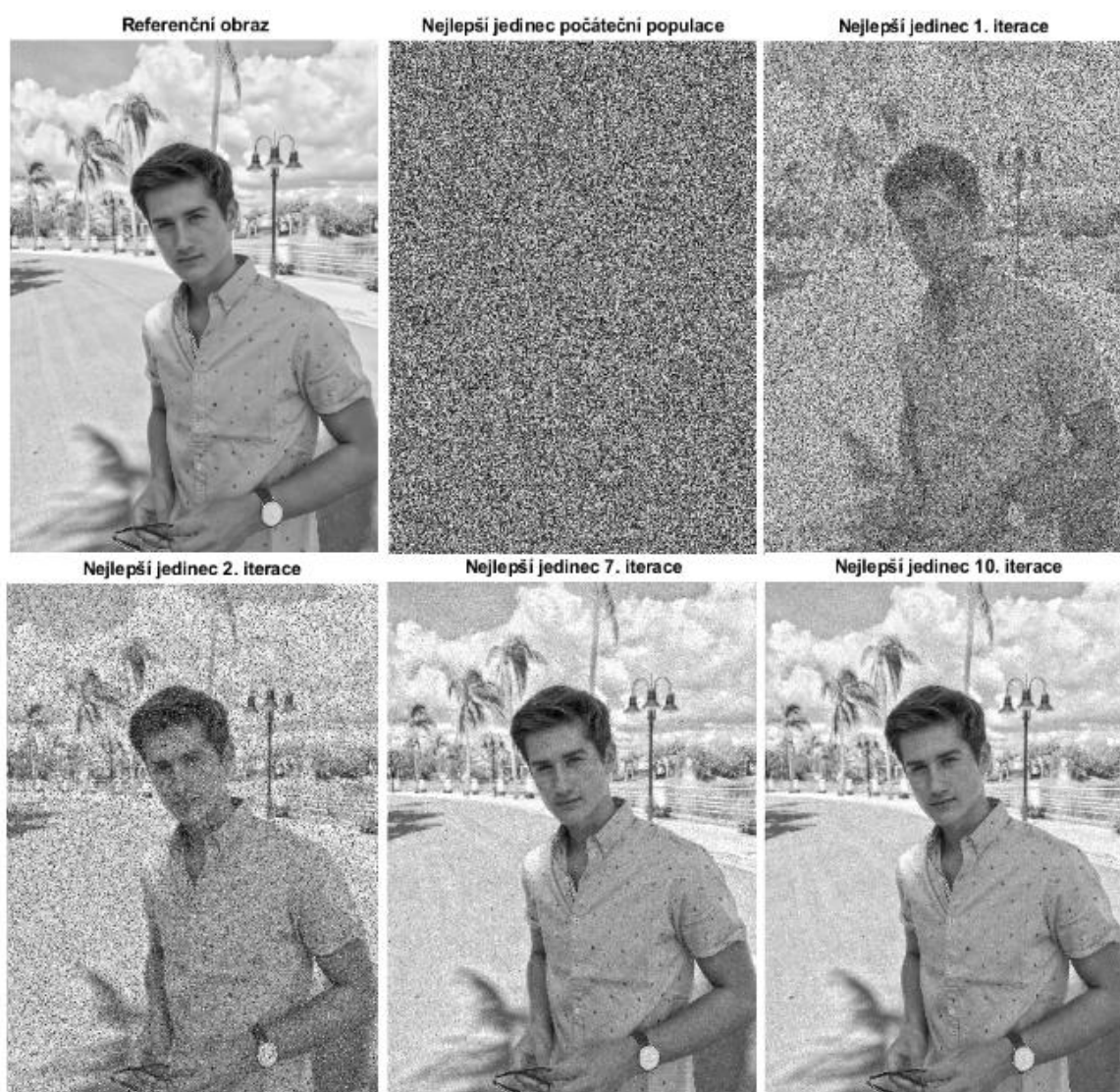
Závislosti kritériální funkce a výpočetního času na počtu jedinců a iterací lze vidět na následujících obrázcích.



Obr. 50: Závislosti kritériální funkce a výpočetního času na počtu jedinců a iterací.

Tabulka 22: Nastavení parametrů algoritmu PSO pro rekonstrukci obrazu.

Parametr	Hodnota
Počet jedinců	4
C_1	2
C_2	2
W	0,5
Počet iterací	10



Obr. 51: Rekonstrukce obrazu algoritmem PSO.

6.4 Srovnání jednotlivých algoritmů

Jednotlivé parametry jsou porovnány na základě kritériální funkce nejlepšího jedince v poslední iteraci a výpočetního času. Ukončovací algoritmus byl zvolen na základě maximálního počtu iterací. Souhrn výsledných parametrů lze vidět v následující tabulce. Testování probíhalo na ilustračním obrazu s rozměry 410x306, a na čtyřech snímcích sítnice z databáze DRIVE s rozměry 565x584. Výsledné parametry jsou dány průměrem získaných parametrů rekonstrukce všech pěti obrazů.

Tabulka 23: Srovnání algoritmů v rekonstrukci obrazu.

	Včelí alg.	Genetický alg.	PSO
Euklidova vzdálenost	49305	113415	325870
Výpočetní čas	10,05 s	22,31 s	13,91 s

Minimální Euklidovu vzdálenost vykazoval včelí algoritmus, který zároveň dosahoval nejnižšího výpočetního času. Nejvyšší hodnoty kritériální funkce dosahoval algoritmus PSO, který byl však proti genetickému algoritmu, jenž vykazoval téměř třikrát lepší hodnoty než PSO, rychlejší. Genetický algoritmus měl ze všech algoritmů po první iteraci nejvyšší připodobnění k referenčnímu obrazu, avšak s dalšími iteracemi dosahoval vyššího připodobnění včelí algoritmus. Genetický algoritmus měl také problémy rekonstruovat pixely, které v referenčním obrázku dosahovaly velmi vysokých, nebo velmi nízkých hodnot. Rozmezí jasových hodnot pixelů je zvoleno na $\langle 0, 255 \rangle$. Jelikož hodnota nových potomků může dosahovat hodnot pouze v rozmezí obou rodičů, je rekonstrukce velmi světlých a velmi tmavých odstínů závislá na počáteční populaci, případně na mutaci jednotlivých pixelů, čímž může dojít k posunu do vyšších, či nižších hodnot.

Zvýšení podobnosti referenčního a rekonstruovaného obrazu by šlo získat zvýšením počtu jedinců a iterací. Parametry byly však voleny způsobem, aby vykazovaly co nejnižší výpočetní čas při dostatečné rekonstrukci obrazu.

Dalším využitím evolučních algoritmů v této oblasti je postupná změna scény jednoho obrazu na druhý. Jediným rozdílem je, že namísto počáteční populace bude přiřazen druhý obraz. V takovém případě je jeden obraz referenční, druhý výchozí a postupem iterací se výchozí obraz mění na referenční. V užším rozměru se tato funkce dá využít jako sdílení části rysů tváří dvou různých lidí. Příklad je uveden na Obr. 52 a Obr. 53. Evoluční algoritmy mohou být také využity v rekonstrukci z projekcí [42], kde autoři použili genetický algoritmus nebo [60], kde byl pro rekonstrukci PET snímků použit fly algoritmus, vykazující mnoho podobností s PSO algoritmem.



Obr. 52: Rekonstrukce výchozího obrazu na referenční (výchozí i referenční obraz převzat z [59]).



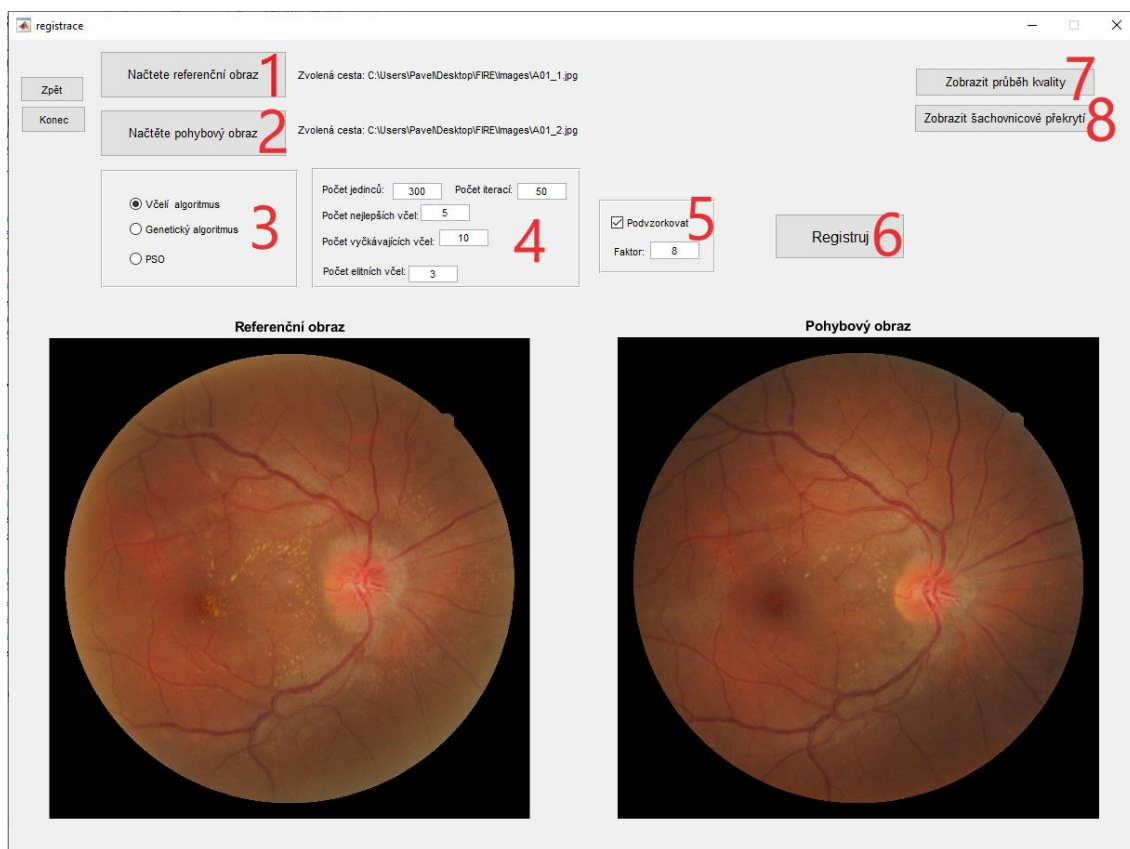
Obr. 53: Rekonstrukce výchozího obrazu na referenční, 2. ukázka (výchozí, resp. referenční obraz převzat z [64], resp. [65]).

7. GRAFICKÉ UŽIVATELSKÉ ROZHRAŇÍ

Pro snazší a pohodlnější manipulaci uživatele při testování jednotlivých algoritmů bylo vytvořeno grafické uživatelské prostředí GUI. Při spuštění programu se objeví hlavní blok, kde si uživatel vybere oblast zpracování obrazu, ve které chce vytvořené algoritmy otestovat. Po kliknutí na požadovanou aplikaci je otevřeno hlavní okno samotné aplikace.

7.1 Registrace obrazu

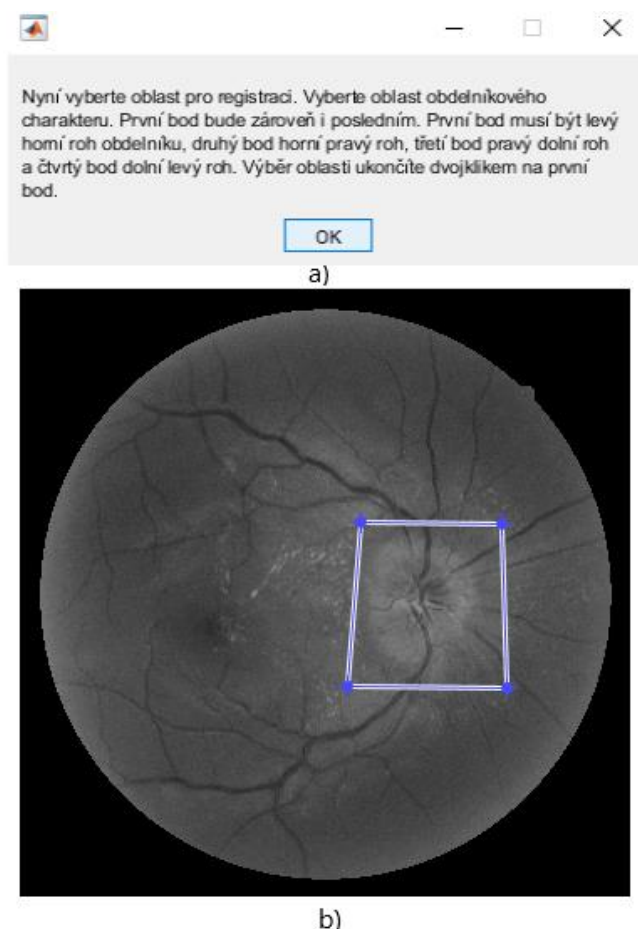
Při výběru registrace obrazu je otevřeno okno na Obr. 54.



Obr. 54: GUI blok – registrace.

Popis hlavních bloků je následovný. Při kliknutí na tlačítko 1 je uživatelem vybrán referenční obraz. Při kliknutí na tlačítko 2 je uživatelem vybrán pohybový obraz. Cesty vybraných souborů jsou zobrazeny vedle korespondujících tlačítek. V bloku 3 si uživatel vybere algoritmus, kterým chce vybrané snímky registrovat. Po výběru algoritmu má uživatel v bloku 4 možnost změnit hodnoty jednotlivých parametrů. Blok 5 uživateli umožňuje podvzorkovat snímky. Tato možnost je především pro snímky větších rozměrů. Defaultně je faktor podvzorkování nastaven na 8, jelikož se předpokládá, že registrace bude probíhat na snímcích z FIRE databáze. Po vybrání referenčního obrazu, pohybového obrazu, algoritmu a jejich parametrů může uživatel spustit registraci. Při stisknutí tlačítka

6 se objeví krátké okno, jenž informuje uživatele, jakým způsobem má vybrat oblast, ze které bude stanovena kritériální funkce obou obrazů, viz Obr. 55 a). Doporučuje se vybrat vysokokontrastní oblast, jako je např. oblast optického disku nebo oblast větvení cév. Po odsouhlasení uživatele tlačítkem OK je zobrazeno okno se samotným výběrem kritériální oblasti. Vhodný výběr oblasti pro výpočet kritériální funkce je zobrazen na Obr. 55. b). Po ukončení registrace je automaticky otevřeno okno s šachovnicovým překryvem registrovaných snímků, toto okno může uživatel zavřít a má možnost zobrazit si průběh kvality průměrného a elitního jedince, toho uživatel využije při stisknutí tlačítka 7. Pro opětovné zobrazení šachovnicového překrytí registrovaných snímků uživatel stiskne tlačítko 8.

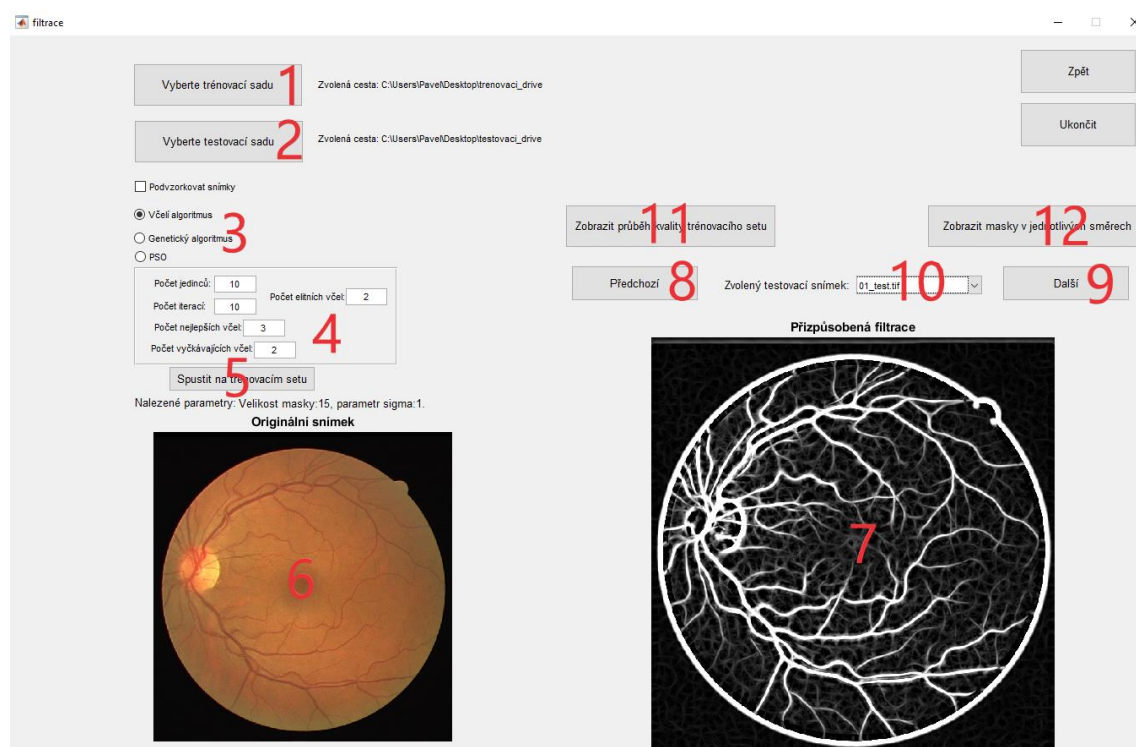


Obr. 55: GUI výběr oblasti pro výpočet kritériální funkce.

7.2 Přizpůsobená filtrace

Při výběru přizpůsobené filtrace je otevřeno okno na Obr. 56. Stisknutím tlačítka 1 je uživatelem vybrána cesta ke složce trénovacího setu. Tlačítko 2 slouží k výběru testovacího setu. Cesty k vybraným složkám jsou zobrazeny vedle korespondujících tlačítek. Složka trénovacího setu musí obsahovat pro jeden snímek celkem tři soubory. Prvním je originální snímek sítnice, druhým je jeho zlatý standard a třetí maska vymezující zorné pole sítnice. Všechny tři snímky musí být pojmenovány tak, aby

následovaly abecedně za sebou, nejjednodušší je, aby začínaly stejným číslem. Originální snímek musí obsahovat koncovku tiff. Zlatý standard a maska zorného pole pak musí obsahovat koncovku giff a platí, že maska zorného pole musí být abecedně dále než zlatý standard. Příklad názvu originálního snímku: 21_training.tiff, zlatý standard: 21.giff, maska zorného pole: 21_training_mask.giff. Ve složce se nesmí kromě těchto snímků vyskytovat žádné další soubory s koncovkami tiff, nebo giff. Pokud jsou dodržena výše uvedená pravidla je program schopen testovat jeden snímek po druhém bez zásahu uživatele a je možné nahrát jakýkoliv trénovací set. Pokud uvedená pravidla dodržena nejsou, je nutné poupravit kód.

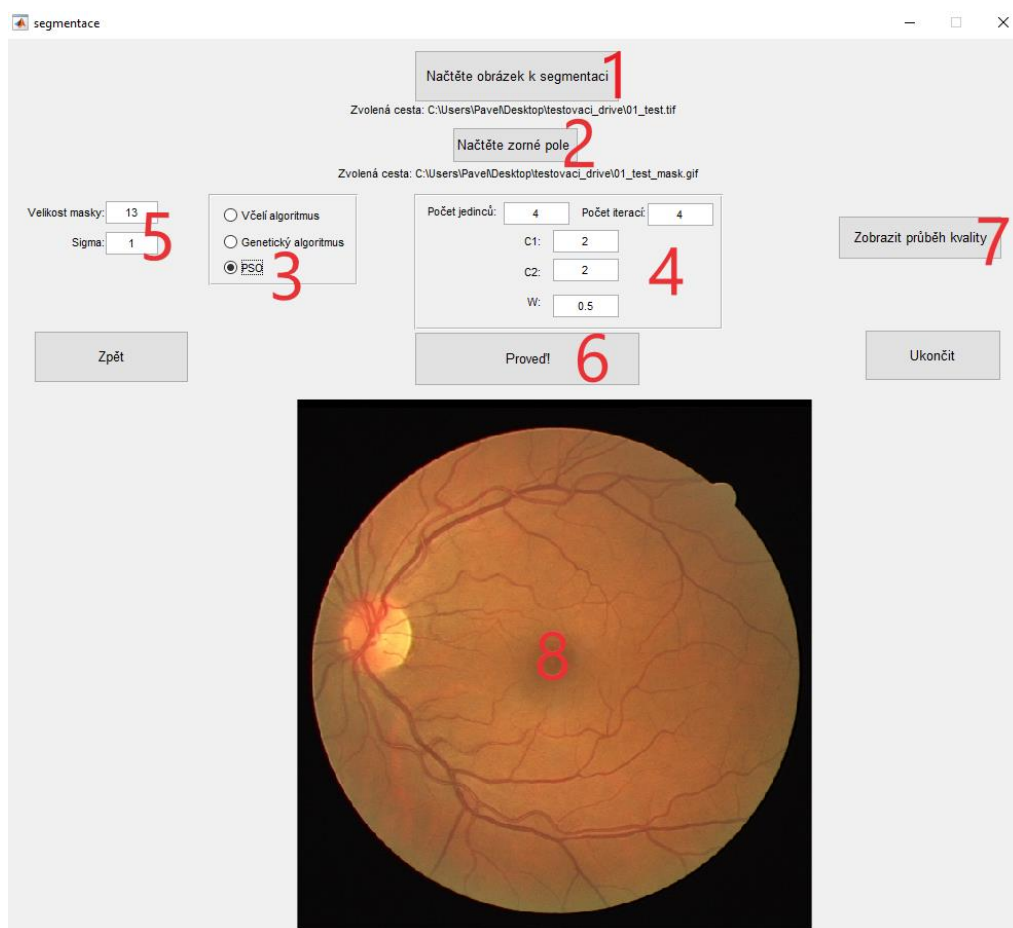


Obr. 56: GUI blok – přizpůsobená filtrace.

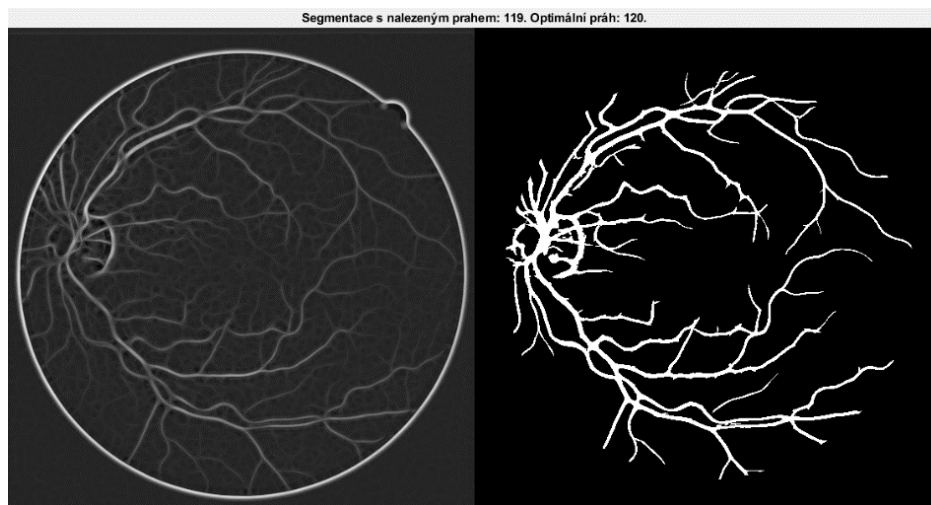
V bloku 3 si uživatel vybere algoritmus, na kterém chce program testovat a v bloku 4 má možnost změnit hodnoty parametrů. Stisknutím tlačítka 5 začne testování. Po ukončení testování se pod blokem 5 zobrazí nalezené parametry a automaticky se otestuje abecedně první snímek testovacího setu. Originální snímek testovacího setu lze vidět v bloku 6 a filtrovaný snímek v bloku 7. Pomocí bloku 8, resp. 9 je uživatel schopen přecházet k předchozím, resp. k následujícím snímkům. V bloku 10 je vyskakovací nabídka, kde jsou přítomny názvy všech snímků testovacího setu. Kliknutím na požadovaný název je snímek filtrován. Uživatel má možnost podívat se na průběh kvality průměrného a elitního jedince kliknutím na tlačítko 11. Tlačítko 12 skrývá jednotlivé masky představující profil cévy.

7.3 Segmentace obrazu

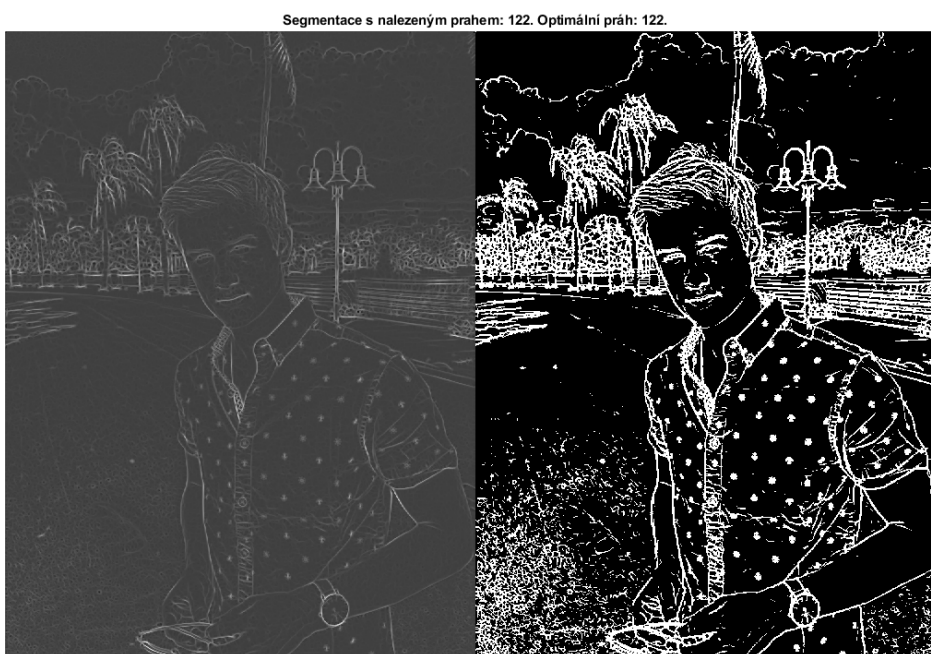
Při výběru segmentace je otevřeno okno na Obr. 57. Tlačítkem 1 uživatel vybere snímek, který má být segmentován a je zobrazen v bloku 8. Tlačítkem 2 je vybrána maska vymezující zorné pole sítnice. Cesty vybraných souborů jsou umístěny pod korespondujícími tlačítky. Jestliže uživatel vynechá výběr masky zorného pole, bude nalezen práh na základě jasových hodnot celého obrazu, čehož se dá využít při segmentaci snímků, jejichž objektem zkoumání není cévní řečiště sítnice, viz Obr 59. V bloku 3 lze vybrat jeden ze tří evolučních algoritmů, při jejichž výběru se objeví jednotlivé parametry, které lze změnit v bloku 4. Blok 5 slouží ke stanovení parametrů masky pro přizpůsobenou filtraci. Tlačítkem 6 je provedeno testování. Po ukončení testování je automaticky otevřeno okno, jež obsahuje fúzi parametrických obrazů a segmentovaný snímek, viz Obr. 58. V titulku tohoto obrázku je rovněž uvedena hodnota nalezeného a optimálního prahu. Tlačítko 7 uživateli umožňuje podívat se na průběh kvality průměrného a elitního jedince.



Obr. 57: GUI blok – segmentace obrazu.



Obr. 58: GUI blok – výsledek segmentace.

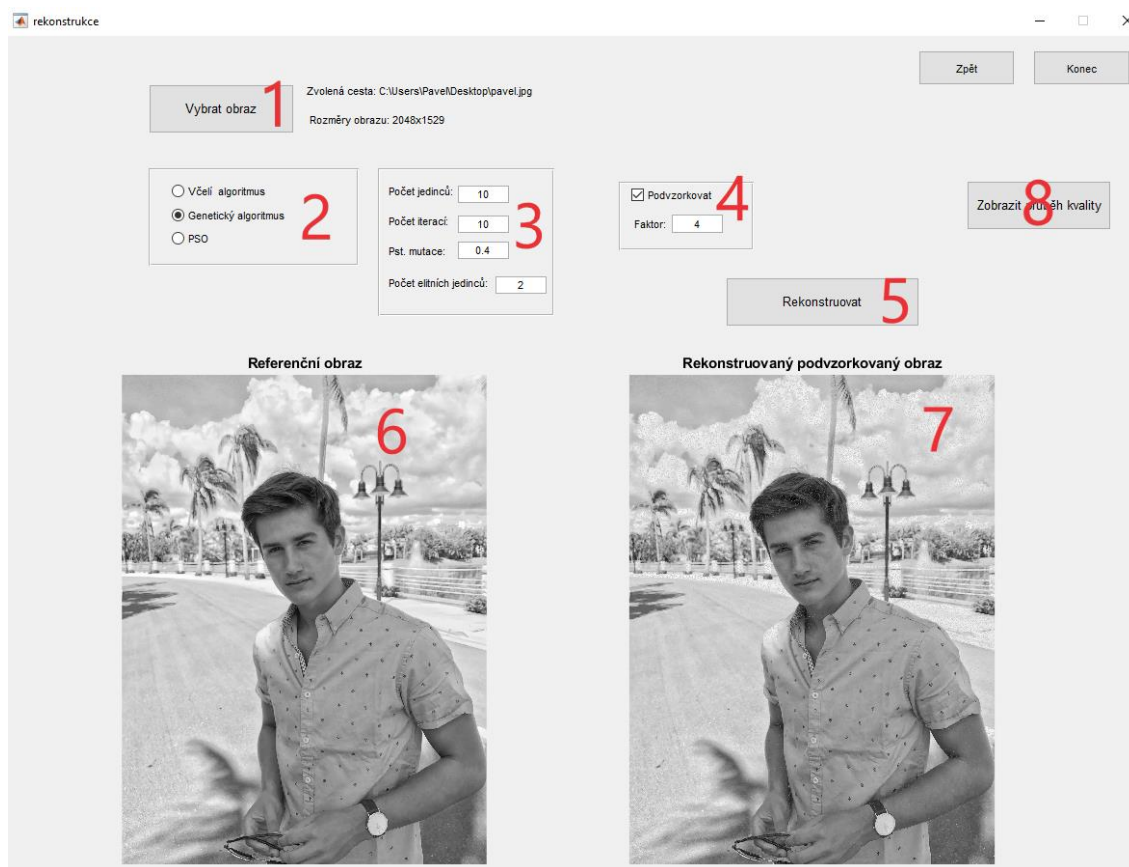


Obr. 59: GUI blok – výsledek segmentace 2. ukázka.

7.4 Rekonstrukce obrazu

Při výběru rekonstrukce obrazu je otevřeno okno na Obr 60. Referenční obraz uživatel vybere kliknutím na tlačítko 1. Obraz lze následně vidět v bloku 6. Cesta a rozměry obrazu jsou umístěny vedle tlačítka 1. Blok 2 slouží k výběru algoritmu pro testování, jehož parametry může uživatel změnit v bloku 3. Blok 4 složí k případnému podvzorkování obrazu, což může několikanásobně urychlit testování. Tlačítkem 5 je spuštěn proces rekonstrukce. V momentě kliknutí je otevřeno nové okno, kde lze vidět

průběh rekonstrukce v jednotlivých iteracích, jako např. na Obr. 51. Po ukončení testování je konečný rekonstruovaný obraz k vidění v bloku 7. Tlačítko 8 slouží k zobrazení průběhu kvality průměrného a elitního jedince.



Obr. 60: GUI blok – rekonstrukce obrazu.

ZÁVĚR

Cílem této diplomové práce bylo seznámit se s využitím optimalizačních technik v obrazových aplikacích. Byly zde popsány vybrané optimalizační metody a aplikace zpracování obrazu. Z optimalizačních metod byly zvoleny tři metody evolučních algoritmů, konkrétně včelí algoritmus, spojitý genetický algoritmus a algoritmus PSO, jež byly detailněji popsány a implementovány v programovacím jazyku Matlab.

Zmíněné evoluční algoritmy byly následně využity v registraci obrazu. Nejprve byly algoritmy otestovány na modelových datech, která byla pro potřebu práce vytvořena. Kriteriační funkcí byla střední kvadratická odchylka. Na těchto modelových datech byla podrobněji popsána optimalizace všech tří evolučních algoritmů, jejichž průběh byl doplněn několika ilustračními obrázky. Algoritmy byly poté otestovány na reálných datech z databáze FIRE, jež obsahuje 134 párů snímků sítnice. Pro jednotlivé algoritmy bylo stanoveno několik možných nastavení s různými parametry. Bylo nalezeno nejvhodnější z nich, které představovalo kompromis vysoké přesnosti a nízké výpočetní náročnosti. Algoritmy byly navzájem srovnány na základě vzdálenosti anotovaných bodů pohybového a referenčního obrazu a výpočetního času. Jednotlivé páry snímků byly také rozřazeny do tří kategorií na základě průměrné odchylky anotovaných bodů obou obrazů. Kategorie A a B obsahovala páry snímků, jejichž přesnost registrace činila menší průměrnou odchylku než 20 pixelů. Kategorie C potom zahrnovala snímky, pro něž registrace nebyla dostatečná a jejichž slícování bylo více nepřesné. Na těchto snímcích mohly algoritmy selhat z důvodu progresu retinopatie, projevující se anatomickými rozdíly, jež způsobily rozdíly intenzit jasu ve snímku. Na snímcích kategorie A a B vykazoval nejpresnější registraci včelí algoritmus, jenž však proti zbylým algoritmům dosahoval téměř dvakrát vyšších hodnot výpočetního času. Nejrychlejším algoritmem byl genetický algoritmus, který však dosahoval nejméně přesných výsledků. Zlepšení přesnosti by šlo docílit předzpracováním jednotlivých snímků nebo použitím kriteriační funkce, jež není založená na porovnání intenzit obou snímků.

Následující kapitola byla věnována optimalizaci parametrů masky pro přizpůsobenou filtraci. Byla využita data z databáze DRIVE, která obsahuje trénovací a testovací set po dvaceti snímcích sítnice. Algoritmy byly použity k nalezení optimálních parametrů pro jedinou masku. Kriteriační funkcí byl podobnostní koeficient Dice mezi jednotlivými snímky a jejich anotovanými daty, která byla společně s maskami zorného pole přiložena ke každému snímku sítnice. Jednotlivými algoritmy byly nalezeny optimální parametry každého snímku trénovacího setu a byl stanoven průměr. Nejlepší výsledky vykazoval včelí algoritmus s nejvyšší hodnotou kriteriační funkce. Mírně nižších hodnot kriteriační funkce potom dosahoval algoritmus PSO, jenž měl téměř dvakrát nižší hodnotu výpočetního času. Mírně vyšší hodnotu výpočetního času algoritmu

PSO měl genetický algoritmus, který vykazoval nejnížší hodnoty kritériální funkce. Přesnější filtrace snímků by byla docílena použitím vyššího počtu masek.

V další kapitole bylo pojednáno o segmentaci obrazu pomocí prahování. Jednotlivé algoritmy byly užity k nalezení optimálního prahu Otsu metodou. Jako vstupní obrazy pro optimalizaci byly využity snímky testovacího setu DRIVE filtrované pomocí masky s parametry nalezenými v předchozí kapitole. Pro každý snímek byla zjištěna hodnota prahu a byl vypočten podobnostní koeficient Dice. Optimální prahy jednotlivých snímků byly nalezeny všemi algoritmy. Dalším předmětem optimalizace bylo nalézt hodnotu prahu, jež bude stejná pro všechny snímky testovacího setu. Kritériální funkcí byl podobnostní koeficient Dice. Včelí algoritmus a PSO našly totožnou hodnotu prahu, pomocí níž byla segmentace úspěšnější než pomocí prahu, jenž byl dosažen genetickým algoritmem. Pro včelí algoritmus a PSO činila senzitivita 72,17 % a specifická 96,28 %. Genetický algoritmus vykazoval senzitivitu 62,54 % a specifickou 96,86 %. PSO algoritmus dosahoval nejnížší hodnoty výpočetního času i nejvyšší hodnoty kritériální funkce, lze tedy prohlásit, že optimalizaci zvládl nejlépe. Ve srovnání s ostatními autory byly uvedené výsledky spíše podprůměrné. Je nutné ovšem zdůraznit, že nalezené parametry masky a hodnota prahu byly totožné pro celý testovací set.

Následující kapitola byla věnována rekonstrukci obrazu, v níž bylo cílem rekonstruovat šedotónové obrazy. Jako v každé zmíněné aplikaci zpracování obrazu, i zde byly nalezeny nejvhodnější parametry algoritmů. Pro správnou činnost rekonstrukce bylo přistoupeno k modifikacím jednotlivých algoritmů. Jednotlivé modifikace byly náležitě popsány pomocí ilustračních obrázků. Jako kritériální funkce byla zvolena Euklidova vzdálenost referenčního a rekonstruovaného obrazu. Nejlepší výsledky vykazoval včelí algoritmus, jenž dosahoval nejnížších hodnot kritériální funkce a nejnížších hodnot výpočetního času. Genetický algoritmus, v porovnání s PSO, dosahoval nižších hodnot kritériální funkce. Výrazně rychleji však pracoval algoritmus PSO.

Pro přehlednější a komfortnější manipulaci při testování bylo vytvořeno grafické uživatelské rozhraní GUI, v němž má uživatel možnost libovolně měnit hodnoty parametrů jednotlivých algoritmů a také mu je dovoleno zvolit si testování jakékoliv z výše uvedené aplikace ve zpracování obrazu.

Kromě aplikací, jež byly implementovány v předložené práci, lze evoluční algoritmy využít také např. ke kompresi obrazu, adaptivní filtraci nebo zvýšení jasu či kontrastu obrazu.

Literatura

- [1] HILL, Derek L G, Philipp G BATCHELOR, Mark HOLDEN a David J HAWKES. Medical image registration. *Physics in Medicine and Biology* [online]. 2001, 46(3), R1-R45 [cit. 2021-5-14]. ISSN 0031-9155. Dostupné z: doi:10.1088/0031-9155/46/3/201
- [2] POLYAK, B.T. Newton's method and its use in optimization. *European Journal of Operational Research* [online]. 2007, 181(3), 1086-1096 [cit. 2021-5-14]. ISSN 03772217. Dostupné z: doi:10.1016/j.ejor.2005.06.076
- [3] JAN, Jiří. Medical image processing, reconstruction and restoration: concepts and methods. Boca Raton: Taylor & Francis, 2006. ISBN isbn0-8247-5849-8.
- [4] ZITOVÁ, Barbara a Jan FLUSSER. Image registration methods: a survey. *Image and Vision Computing* [online]. 2003, 21(11), 977-1000 [cit. 2021-5-14]. ISSN 02628856. Dostupné z: doi:10.1016/S0262-8856(03)00137-9
- [5] ZAGORCHEV, L. a A. GOSHTASBY. A comparative study of transformation functions for nonrigid image registration. *IEEE Transactions on Image Processing* [online]. 2006, 15(3), 529-538 [cit. 2021-5-14]. ISSN 1057-7149. Dostupné z: doi:10.1109/TIP.2005.863114
- [6] MODERSITZKI, Jan: FAIR: flexible algorithms for image registration. Philadelphia: Society for Industrial and Applied Mathematics, 2009, ISBN 978-0-898716-90-0, 189 s.
- [7] SIMON, Dan. Evolutionary optimization algorithms: biologically-inspired and population-based approaches to computer intelligence. Wiley: Hoboken, [2013]. ISBN 978-0-470-93741-9.
- [8] Ruder, S. (2016). An overview of gradient descent optimization algorithms. NUI Galway: Insight center for data analytics, (cite arxiv:1609.04747).
- [9] CERDÀ, Víctor, Juan Luis CERDÀ a Abubakr M. IDRIS. Optimization using the gradient and simplex methods. *Talanta* [online]. 2016, 148, 641-648 [cit. 2021-5-14]. ISSN 00399140. Dostupné z: doi:10.1016/j.talanta.2015.05.061

- [10] TVRDÍK, J. Evoluční algoritmy, Ostravská univerzita, 2004, 73 s.
- [11] GOFFE, William L., Gary D. FERRIER a John ROGERS. Global optimization of statistical functions with simulated annealing. *Journal of Econometrics* [online]. 1994, 60(1-2), 65-99 [cit. 2021-01-05]. ISSN 03044076.
- [12] KOZUMPLÍK J. Evoluční algoritmy (MPC-EAL): Evoluční algoritmy 3. Elektronická přednáška FEKT VUT v Brně. 2020, <https://www.vutbr.cz/elearning/>
- [13] LÉVY VÉHEL, Jacques, Franklin MENDIVIL a Evelyne LUTTON. Overcompressing JPEG Images with Evolution Algorithms. GIACOBINI, Mario, ed. *Applications of Evolutionary Computing* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, s. 383-390 [cit. 2021-01-01]. Lecture Notes in Computer Science. ISBN 978-3-540-71804-8.
- [14] Petr Pošík. Evoluční algoritmy s reálnou prezentací. Evoluční strategie. Rojová inteligence. Optimalizace rojením částic. Mravenčí kolonie [online prezentace]. Praha: ČVUT, [cit. 2020-12-12]. Dostupné z: https://cw.fel.cvut.cz/old/_media/courses/y33au/09swarm.pdf.
- [15] PHAM, D.T., A. GHANBARZADEH, E. KOÇ, S. OTRI, S. RAHIM a M. ZAIDI. The Bees Algorithm — A Novel Tool for Complex Optimisation Problems. *Intelligent Production Machines and Systems* [online]. Elsevier, 2006, 2006, s. 454-459 [cit. 2021-5-14]. ISBN 9780080451572. Dostupné z: doi:10.1016/B978-008045157-2/50081-X
- [16] Digital Image Interpolation. In: Cambridge in Colour, [online], [cit. 2020-11-29]. Dostupné z: <https://www.cambridgeincolour.com/tutorials/image-interpolation.htm>.
- [17] KOLÁŘOVÁ, Jana, RONKOVÁ, Pavla. Kriteriaální funkce pro registraci obrazů sítnice. Brno: Fakulta elektrotechniky a komunikačních technologií VUT v Brně. 2013, volume 15, issue 4. ISSN 1213-1539.
- [18] NAG, Sayan. Vector quantization using the improved differential evolution algorithm for image compression. *Genetic Programming and Evolvable Machines* [online]. 2019, 20(2), 187-212 [cit. 2021-5-14]. ISSN 1389-2576. Dostupné z: doi:10.1007/s10710-019-09342-8
- [19] BHANDARKAR, S.M. a HUI ZHANG. Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation* [online]. 3(1), 1-21 [cit. 2021-5-14]. ISSN 1089778X. Dostupné z: doi:10.1109/4235.752917

- [20] CHALOTRA, Sherry, Sumeet Kaur SEHRA a Sukhjit Singh SEHRA. A systematic review of applications of Bee Colony Optimization. In: *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)* [online]. IEEE, 2016, 2016, s. 257-260 [cit. 2021-5-14]. ISBN 978-1-5090-2084-3. Dostupné z: doi:10.1109/ICICCS.2016.7542297

- [21] HAO, Wenrui. A gradient descent method for solving a system of nonlinear equations. *Applied Mathematics Letters* [online]. 2021, 112 [cit. 2021-5-14]. ISSN 08939659. Dostupné z: doi:10.1016/j.aml.2020.106739

- [22] SLOWIK, Adam a Halina KWASNICKA. Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications* [online]. 2020, 32(16), 12363-12379 [cit. 2021-5-14]. ISSN 0941-0643. Dostupné z: doi:10.1007/s00521-020-04832-8

- [23] CHELOUAH, R. a P. SIARRY. A continuous Genetic Algorithm Designed for the Global Optimization of Multimodal Functions. *Journal of Heuristics*. 6(2), 191-213 [cit. 2021-5-14]. ISSN 13811231. Dostupné z: doi:10.1023/A:1009626110229

- [24] LIBERTI, Leo, Carlile LAVOR, Nelson MACULAN a Antonio MUCHERINO. Euclidean Distance Geometry and Applications. *SIAM Review* [online]. 2014, 56(1), 3-69 [cit. 2021-5-14]. ISSN 0036-1445. Dostupné z: doi:10.1137/120875909

- [25] KOZUMPLÍK J. Evoluční algoritmy (MPC-EAL): Evoluční algoritmy 5. Elektronická přednáška FEKT VUT v Brně. 2020, <https://www.vutbr.cz/elearning/>

- [26] HAUPT, Randy L. a S. E. HAUPT. *Practical genetic algorithms*. 2nd ed. Hoboken: John Wiley, 2004. ISBN 0-471-45565-2.

- [27] DAGDIA, Zaineb Chelly a Miroslav MIRCHEV. When Evolutionary Computing Meets Astro- and Geoinformatics. *Knowledge Discovery in Big Data from Astronomy and Earth Observation* [online]. Elsevier, 2020, 2020, s. 283-306 [cit. 2021-4-28]. ISBN 9780128191545. Dostupné z: doi:10.1016/B978-0-12-819154-5.00026-6

- [28] SEIXAS GOMES DE ALMEIDA, Bruno a Victor COPPO LEITE. Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems. DEL SER, Javier, Esther VILLAR a Eneko OSABA, ed. *Swarm Intelligence - Recent Advances, New Perspectives and Applications* [online]. IntechOpen, 2019, 4. 12. 2019 [cit. 2021-4-28]. ISBN 978-1-78984-536-5. Dostupné z: doi:10.5772/intechopen.89633

- [29] ENGELBRECHT, Andries. Particle swarm optimization: Velocity initialization. In: *2012 IEEE Congress on Evolutionary Computation* [online]. IEEE, 2012, 2012, s. 1-8 [cit. 2021-4-29]. ISBN 978-1-4673-1509-8. Dostupné z: doi:10.1109/CEC.2012.6256112
- [30] JAYAPRAKASAM, S., S.K.A. RAHIM a Chee Yen LEOW. PSOGSA-Explore: A new hybrid metaheuristic approach for beampattern optimization in collaborative beamforming. *Applied Soft Computing* [online]. 2015, 30, 229-237 [cit. 2021-4-29]. ISSN 15684946. Dostupné z: doi:10.1016/j.asoc.2015.01.024
- [31] SAHAB, Mohammed Ghasem, Vassili V. TOROPOV a Amir Hossein GANDOMI. A Review on Traditional and Modern Structural Optimization. *Metaheuristic Applications in Structures and Infrastructures* [online]. Elsevier, 2013, 2013, s. 25-47 [cit. 2021-4-29]. ISBN 9780123983640. Dostupné z: doi:10.1016/B978-0-12-398364-0.00002-4
- [32] ZHONG, Jin a Hao WU. Evolutionary Game Algorithm for Image Segmentation. *Journal of Electrical and Computer Engineering* [online]. 2017, 2017, 1-10 [cit. 2021-4-29]. ISSN 2090-0147. Dostupné z: doi:10.1155/2017/8746010
- [33] BHANDARKAR, S.M. a HUI ZHANG. Image segmentation using evolutionary computation. *IEEE Transactions on Evolutionary Computation* [online]. 3(1), 1-21 [cit. 2021-4-29]. ISSN 1089778X. Dostupné z: doi:10.1109/4235.752917
- [34] PAL, Nikhil R a Sankar K PAL. A review on image segmentation techniques. *Pattern Recognition* [online]. 1993, 26(9), 1277-1294 [cit. 2021-4-29]. ISSN 00313203. Dostupné z: doi:10.1016/0031-3203(93)90135-J
- [35] HUANG, Mengxing, Wenjiao YU a Donghai ZHU. An Improved Image Segmentation Algorithm Based on the Otsu Method. In: *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* [online]. IEEE, 2012, 2012, s. 135-139 [cit. 2021-4-29]. ISBN 978-1-4673-2120-4. Dostupné z: doi:10.1109/SNPD.2012.26
- [36] MERZOUGUI, Mohammed a Ahmad El ALLAOUI. Region growing segmentation optimized by evolutionary approach and Maximum Entropy. *Procedia Computer Science* [online]. 2019, 151, 1046-1051 [cit. 2021-4-30]. ISSN 18770509. Dostupné z: doi:10.1016/j.procs.2019.04.148
- [37] POHLE, Regina, Klaus D. TOENNIES, Milan SONKA a Kenneth M. HANSON. [online]. In: . 3. 7. 2001, s. 1337-1346 [cit. 2021-4-30]. Dostupné z: doi:10.1117/12.431013

- [38] GRAU, V., A.U.J. MEWES, M. ALCANIZ, R. KIKINIS a S.K. WARFIELD. Improved Watershed Transform for Medical Image Segmentation Using Prior Information. *IEEE Transactions on Medical Imaging* [online]. 2004, 23(4), 447-458 [cit. 2021-4-30]. ISSN 0278-0062. Dostupné z: doi:10.1109/TMI.2004.824224
- [39] Jeyavathana, R., Balasubramanian, D.R., & Pandian, A. (2016). A Survey: Analysis on Pre-processing and Segmentation Techniques for Medical Images. ISSN 2321-2705.
- [40] Wallingford R.M., Siwek E.M., Gray J.N. (1992) Application of Two-Dimensional Matched Filters to X-Ray Radiographic Flaw Detection and Enhancement. In: Thompson D.O., Chimenti D.E. (eds) Review of Progress in Quantitative Nondestructive Evaluation. Advances in Cryogenic Engineering, vol 28. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-3344-3_113
- [41] JAN, Jiří. Medical image processing, reconstruction and restoration: concepts and methods. Boca Raton: Taylor & Francis, 2006. ISBN 0-8247-5849-8.
- [42] KODALI, Shyam P., Sunith BANDARU, K. DEB, Prabhat MUNSHI a N. N. KISHORE. Applicability of genetic algorithms to reconstruction of projected data from ultrasonic tomography. In: *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08* [online]. New York, New York, USA: ACM Press, 2008, 2008, s. 1705- [cit. 2021-5-10]. ISBN 9781605581309. Dostupné z: doi:10.1145/1389095.1389419
- [43] AIT-AOUDIA, Samy, El-Hachemi GUERROUT a Ramdane MAHIOU. Medical Image Segmentation Using Particle Swarm Optimization. In: *2014 18th International Conference on Information Visualisation* [online]. IEEE, 2014, 2014, s. 287-291 [cit. 2021-4-30]. ISBN 978-1-4799-4103-2. Dostupné z: doi:10.1109/IV.2014.68
- [44] SHETA, Alaa, Malik S. BRAIK a Sultan ALJAHDAI. Genetic Algorithms: A tool for image segmentation. In: *2012 International Conference on Multimedia Computing and Systems* [online]. IEEE, 2012, 2012, s. 84-90 [cit. 2021-4-30]. ISBN 978-1-4673-1520-3. Dostupné z: doi:10.1109/ICMCS.2012.6320144
- [45] TAO, Wenbing, Hai JIN a Liman LIU. Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognition Letters* [online]. 2007, 28(7), 788-796 [cit. 2021-4-30]. ISSN 01678655. Dostupné z: doi:10.1016/j.patrec.2006.11.007

- [46] AL-RAWI, Mohammed a Huda KARAJEH. Genetic algorithm matched filter optimization for automated detection of blood vessels from digital retinal images. *Computer Methods and Programs in Biomedicine* [online]. 2007, 87(3), 248-253 [cit. 2021-4-30]. ISSN 01692607. Dostupné z: doi:10.1016/j.cmpb.2007.05.012

- [47] SREEJINI, K.S. a V.K. GOVINDAN. Improved multiscale matched filter for retina vessel segmentation using PSO algorithm. *Egyptian Informatics Journal* [online]. 2015, 16(3), 253-260 [cit. 2021-4-30]. ISSN 11108665. Dostupné z: doi:10.1016/j.eij.2015.06.004

- [48] AHMAD, Yasmin Mumtaz, Shahnorbanun SAHRAN, Afzan ADAM a - SYAZARINA. Linear Intensity-Based Image Registration. *International Journal of Advanced Computer Science and Applications* [online]. 2018, 9(12) [cit. 2021-5-1]. ISSN 21565570. Dostupné z: doi:10.14569/IJACSA.2018.091231

- [49] YUCE, Baris, Michael PACKIANATHER, Ernesto MASTROCINQUE, Duc PHAM a Alfredo LAMBIASE. Honey Bees Inspired Optimization Method: The Bees Algorithm. *Insects* [online]. 2013, 4(4), 646-662 [cit. 2021-5-2]. ISSN 2075-4450. Dostupné z: doi:10.3390/insects4040646

- [50] ZIARATI, Koorush, Reza AKBARI a Vahid ZEIGHAMI. On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing* [online]. 2011, 11(4), 3720-3733 [cit. 2021-5-2]. ISSN 15684946. Dostupné z: doi:10.1016/j.asoc.2011.02.002

- [51] SARVAMANGALA, D. R. a Raghavendra V. KULKARNI. Swarm Intelligence Algorithms for Medical Image Registration: A Comparative Study. MANDAL, J. K., Paramartha DUTTA a Somnath MUKHOPADHYAY, ed. *Computational Intelligence, Communications, and Business Analytics* [online]. Singapore: Springer Singapore, 2017, 2017-09-26, s. 451-465 [cit. 2021-5-3]. Communications in Computer and Information Science. ISBN 978-981-10-6429-6. Dostupné z: doi:10.1007/978-981-10-6430-2_35

- [52] CHANWIMALUANG, T. a GUOLIANG FAN. An efficient blood vessel detection algorithm for retinal images using local entropy thresholding. In: *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03* [online]. IEEE, 2003, V-21-V-24 [cit. 2021-5-4]. ISBN 0-7803-7761-3. Dostupné z: doi:10.1109/ISCAS.2003.1206162.

- [53] ODSTRČILÍK, J. Analýza barevných snímků sítnice se zaměřením na segmentaci cévního řečiště: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 82 s., 2 přílohy. Vedoucí diplomové práce je prof. Ing. Jiří Jan, CSc.
- [54] ODSTRČILÍK, J., J. JAN, J. GAZÁREK a R. KOLÁŘ. Improvement of Vessel Segmentation by Matched Filtering in Colour Retinal Images. DÖSSEL, Olaf a Wolfgang C. SCHLEGEL, ed. *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, 2009, s. 327-330 [cit. 2021-5-4]. IFMBE Proceedings. ISBN 978-3-642-03890-7. Dostupné z: doi:10.1007/978-3-642-03891-4_87
- [55] HASSAN, Gehad, Nashwa EL-BENDARY, Aboul Ella HASSANIEN, Ali FAHMY, Shueb ABULLAH M. a Vaclav SNASEL. Retinal Blood Vessel Segmentation Approach Based on Mathematical Morphology. *Procedia Computer Science* [online]. 2015, 65, 612-622 [cit. 2021-5-7]. ISSN 18770509. Dostupné z: doi:10.1016/j.procs.2015.09.005
- [56] MARÍN, D, A AQUINO, M E GEGUNDEZ-ARIAS a J M BRAVO. A New Supervised Method for Blood Vessel Segmentation in Retinal Images by Using Gray-Level and Moment Invariants-Based Features. *IEEE Transactions on Medical Imaging* [online]. 2011, 30(1), 146-158 [cit. 2021-5-7]. ISSN 0278-0062. Dostupné z: doi:10.1109/TMI.2010.2064333
- [57] VLACHOS, Marios a Evangelos DERMATAS. Multi-scale retinal vessel segmentation using line tracking. *Computerized Medical Imaging and Graphics* [online]. 2010, 34(3), 213-227 [cit. 2021-5-7]. ISSN 08956111. Dostupné z: doi:10.1016/j.compmedimag.2009.09.006
- [58] Seyedali Mirjalili (2021). Binary Image Reconstruction Using The Genetic Algorithm (<https://www.mathworks.com/matlabcentral/fileexchange/67695-binary-image-reconstruction-using-the-genetic-algorithm>), MATLAB Central File Exchange. Retrieved May 1, 2021.
- [59] Goal, 2020 [online]. *Barcelona's 2020-21 kit: New home and away jersey styles and release dates*. [cit 5.5.2021]. Dostupné z: <https://www.goal.com/en/news/barcelonas-2020-21-kit-new-home-and-away-jersey-styles-and/xspq7soikd9t1dz052mmfcfh4>
- [60] KODALI, Shyam P., Sunith BANDARU, K. DEB, Prabhat MUNSHI a N. N. KISHORE. Applicability of genetic algorithms to reconstruction of projected data from ultrasonic tomography. In: *Proceedings of the 10th annual conference on Genetic and evolutionary*

- computation - GECCO '08* [online]. New York, New York, USA: ACM Press, 2008, 2008, s. 1705- [cit. 2021-5-10]. ISBN 9781605581309. Dostupné z: doi:10.1145/1389095.1389419
- [61] XIONG, Shengwu, Yair CENSOR, Mingyue DING a Jinglei GUO. *Evolutionary algorithm for compression of gray-scale images* [online]. In: . 2001-9-25, s. 392- [cit. 2021-5-10]. Dostupné z: doi:10.1117/12.441612
- [62] WHITE, Michael S. a Stuart J. FLOCKTON. Adaptive Recursive Filtering Using Evolutionary Algorithms. DASGUPTA, Dipankar a Zbigniew MICHALEWICZ, ed. *Evolutionary Algorithms in Engineering Applications* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, 1997, s. 361-376 [cit. 2021-5-10]. ISBN 978-3-642-08282-5. Dostupné z: doi:10.1007/978-3-662-03423-1_21
- [63] SURESH, Shilpa a Shyam LAL. Modified differential evolution algorithm for contrast and brightness enhancement of satellite images. *Applied Soft Computing* [online]. 2017, 61, 622-641 [cit. 2021-5-10]. ISSN 15684946. Dostupné z: doi:10.1016/j.asoc.2017.08.019
- [64] Cool Walls, 2019 [online]. *Image ID:6097617/1080x1920 fox, animals, hd for iphone 6, 7, 8 wallpaper*. [cit 5.5.2021]. Dostupné z: <https://coolwallpapers.me/6097617-fox-animals-hd.html>
- [65] Wallpaper-house [online]. *1080x1920 Animal On Tree Wallpapers HD*. [cit 5.5.2021]. Dostupné z: <https://wallpaper-house.com/wallpaper-id-268014.php>